COMMODORE & AMIGA nr 6



NR INDEKSU 355216
ISSN 0867-8022

Cena 10 000 zł

czerwiec 1992 r.

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW «COMMODORE»

- C-64 kontra ATARI 800XL
- RAPORTY — AMIGA:
  - Zewnętrzne stacje firmy TOMS
  - Rozszerzenie RAM do 2.5 MB
  - Action Replay MK III
- LC-20 i polskie znaki
- Od kuchni: układ VIC i AGNUS
- PROGRAMOTEKA:
  - 64 duszki,
  - Kulki,
  - TURBO dla C-16/116 i PLUS4

... i wiele innych!

Jak zapewne Czytelnicy pamiętają, w numerze „C&A" 04/92 zamieściliśmy ankietę. Choć jeszcze nie mogę opublikować wyników ostatecznych muszę przyznać, że cała redakcja jest pod wrażeniem Waszej reakcji. W chwili gdy to piszę jest środek maja, a liczba nadesłanych ankiet stanowi już ponad 4% nakładu. To dużo, nawet bardzo dużo i szczerze mówiąc nie spodziewałem się nawet tak żywiołowego i gorącego odzewu z Waszej strony.

Wierzę, że cała ankieta zostanie opracowana do końca sierpnia br.; losowanie nagród odbędzie się na pewno w sierpniu, a wyniki opublikujemy w numerze wrześniowym. Dziękuję Wam również za wiele życzeń i serdeczne listy jakie otrzymujemy wraz z ankietami.

Skoro jestem już przy korespondencji: posiadacze Amigi zarzucają nam, że poświęcamy ich komputerom zbyt mało miejsca, to samo robią użytkownicy C-64. Często proponujecie również podział na dwa czasopisma tematyczne. Wierzcie mi, że sam chętnie rozszerzyłbym „C&A" co najmniej dwukrotnie, jednak nie zależy to tylko ode mnie. Co do uruchomienia nowego tytułu za wcześnie by się w tej sprawie wypowiadać. W każdym bądź razie uważnie analizujemy rynek; potencjalnie takie możliwości już istnieją.

Od czerwca rozpoczynamy (na razie na próbę) sprzedaż kaset magnetofonowych zawierających głównie programy opublikowane w „C&A" oraz programy public domain. Będzie tam również kilka niespodzianek — pełną listę programów, cenę oraz specjalny kupon znajdziesz w tym numerze „C&A". Ten sam kupon może również posłużyć do zamówienia pierwszej dyskietki z programami dla Amigi. Znajdują się na niej pliki dla drukarki LC-20 oraz (między innymi) moduły dla Protrackera. Dodam jeszcze, że to nie koniec niespodzianek z dyskietkami i kasetami — do tego tematu niebawem powrócę.

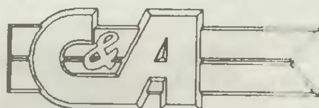
Nadal będzie trwała dobra passa dla majsterkowiczów. Nasi redakcyjni elektronicy opracowali kilka ciekawych urządzeń tak dla C-64 jak i dla Amigi.

Przed nami zaś wakacje i egzaminy. Wraz z całą redakcją życzę Wam połamania pióra i — do zobaczenia za miesiąc.

KLAUDIUSZ DYBOWSKI

M·E·N·U

magazyn użytkowników komputerów «COMMODORE»



Redaktor naczelny: KLAUDIUSZ DYBOWSKI
Sekretarz redakcji: CHRISTIAN GRZENKOWICZ
Opracowanie graficzne: JOLANTA PRZEŹDZIECKA
Redaktor techniczny: JOLANTA SZUMOWSKA
Zdjęcia: JERZY STOKOWSKI

Stali współpracownicy: ANDRZEJ BOBEK (szef Działu Amigi), BARTŁOMIEJ DRAMCZYK, JERZY DUDEK, MARIUSZ FERDYN, BARTŁOMIEJ KAZIMIERCZAK, PIOTR LISZEWSKI, RAFAŁ PIASEK, BARTOSZ SMAGA, RAFAŁ WIOSNA

Redakcja: ul. Wasilkowskiego 7, 02-776 WARSZAWA tel./BBS: 643-1840
Kontakt z Czytelnikami: pon-pt w godzinach 1000-1700
Wydawca: Spółdzielnia „Bajtek" ul. Wspólna 61 00-687 Warszawa tel./fax: 21-12-05
Skład i druk: Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF", Sp. Akc. Ciechanów
Korekta:
Nr zlecenia:
Nakład: 72 tys. egzemplarzy

KTO SIĘ BAWI AMIGĄ

Wśród listów, które dostałem z listy I-Amiga był również ten, który, jak sądzę, będzie "kulami do armat" dla poszukujących argumentów do zaciekłej dyskusji na temat przewagi ich "przyjaciółki" nad innymi komputerami. Dla innych będzie to prawdopodobnie informacją o profesjonalnych zastosowaniach Amigi. A oto lista jej komercyjnych użytkowników:

☐ Lucasfilms Graphic Co. używa Amigi do wszystkich prac związanych z przygotowaniem produkcji.

☐ Disney Word używa setek Amig do sterowania i kontroli animacji.

☐ Ponad 800 stacji telewizji kablowej wykorzystuje Amigi do produkcji tzw. czołówek ekranów tytułowych, nakładania tekstów i innych prac.

☐ CNN Weather, wraz z wieloma lokalnymi stacjami meteo jest całkowicie przygotowywana na Amidze.

☐ Artyści estradowi tacy jak B.B. King i Frank Zappa twierdzą, że jest bezcenna.

☐ Wyse Advertising (firma reklamowa) projektuje prospekty reklamowe i plakaty na Amidze.

☐ Departament policji w Houston używa Amigi do produkcji instruktażowych filmów video.

☐ Elektroniczne tablice wyników na stadionie Joe Robbiego w Miami są sterowane Amigą.

☐ Oprogramowanie do popularnego (na zachodzie) systemu gier Lynx jest opracowywane na Amidze.

☐ Amigi są szeroko stosowane przez Ohio State University, Harvard, MIT, UCLA (to tylko kilka bardziej znanych).

☐ Prezentacja Atlanty Narodowemu Komitetowi Olimpijskiemu była wykonana na Amidze. (IBM PS/2 NEXT i Macintosh nie mogły ponoć podołać temu zadaniu).

☐ Studia Universal produkują na Amigach rysunki i animacje video.

☐ W stanie Alberta (Kanada) Zarząd Telekomunikacji używa około 600 Amig do koordynacji połączeń telefonicznych.

☐ Dowództwo Obrony Strategicznej przeprowadza symulacje ataku i obrony rakietowej na Amigach.

☐ Szkoła medyczna UCLA używa Amigi do badań nad ruchem człowieka oraz stosuje różne techniki wizyjne do zobrazowania ludzkiego mózgu.

☐ Smith-Kettlewel Eye Reasearch Institute w San Francisco używa Amig w badaniach wzroku i nad wzrokiem oraz przyrządami okulistycznymi.

☐ W Komisji Energi Atomowej Amigi służą jako inteligentne terminale superkomputera CRAY.

☐ NBC (stacja radiowa) Amigą tworzy okładki do swoich książek z serii Amazing Stories, a ABC (stacja radiowa) wykonuje tło do komiksów Max Headroom.

☐ Nawet Apple Computer stosuje je wraz z systemem slajdów do prezentacji osiągnięć przed radą nadzorczą.

☐ Louisiana Power and Light stosuje Amigi w sieci poczty elektronicznej do szybkiego przekazywania wiadomości swoim pracownikom.

☐ Departament Weteranów używa Amigi jako podstawy swojego programu rekonwalescencji środkami wizyjnymi w szpitalach.

☐ Metroscan Traffic używa Amig do generowania animowanych diagramów o nasileniu ruchu dla stacji telewizyjnych podających poran- ne wiadomości.

☐ Lotniska międzynarodowe Boston Logan i Toronto Pearson będą instalowały Amigi w celu uruchomienia terminali informacyjnych.

☐ Todd Rundgeren używa 10 Amig 2500 wypo- sażonych w Video Toastery i 300 MB twarde dyski do produkcji video klipów do swojego programu "Change Myself".

☐ Około 5 holenderskich oddziałów policji używa Amig do obsługi wewnętrznej sieci łączności.

☐ Z Amigi korzysta również Roy Clay, który zebrał te informacje; użwam jej również ja, Paweł Galas, czyli autor tego opracowania.

☐ W wielkiej Brytanii Amig używa też policja brytyjska do ewidencjonowania niesfornmych kibiców.

☐ Z Amig mają też korzystać inżynierowie firmy Rolls-Royce posługujący się nimi podczas projektowania CAD.



ZEWNĘTRZNA STACJA DYSKÓW do AMIGI

Praca na Amidze z jedną stacją dysków jest co najmniej trudna, a w niektórych programach w ogóle niemożliwa. Istnieją dwie możliwości rozwiązania tego problemu: kupno gotowej stacji lub podłączenie napędu od komputera IBM. To drugie rozwiązanie jest dużo tańsze.

Złącze stacji dysków w Amidze jest kompatybilne ze złączem Shugart stosowanym w popularnych „pecetach". Umożliwia to podłączenie w prosty sposób do czterech napędów (3.5" lub 5.25"). Napędy są wybierane (uaktywniane) jednym z czterech sygnałów SELx, gdzie x jest numerem wybranego napędu. Ponieważ Amiga ma już jeden wbudowany napęd, tylko linie SEL1, SEL2 i SEL3 są dostępne na gnieździe stacji zewnętrznych. Linia SEL0 jest podłączona do złącza wewnętrznej stacji dysków. Przed rozpoczęciem podłączania napędów proponuję zapoznać się z funkcjami poszczególnych sygnałów złącza Shugart:

SELx
Amiga używa tych linii do wyboru jednego z czterech napędów. Za wyjątkiem linii MTRX i DRES wszystkie pozostałe sygnały są aktywne dopiero po ustawieniu jednego z sygnałów SEL.

MTRX
Normalnie linia ta jest używana do włączania silników wszystkich podłączonych napędów. Ponieważ jest ona podłączona do każdego z napędów, jej uaktywnienie spowoduje uruchomienie silników wszystkich napędów jednocześnie. Jest to rozwiązanie niedopuszczalne, dlatego też konieczne jest zastosowanie dla każdego z napędów prostego przerzutnika sprawdzającego stan linii MTRX w momencie uaktywnienia sygnału SEL odpowiadającego danej stacji. Wyjście przerzutnika jest podłączone do linii MTR w napędzie. Umożliwia to niezależne włączanie i wyłączanie silników w napędach. Dla przykładu: jeśli linia SEL0 zostanie ustawiona w stan aktywny (w tym przypadku jest to stan niski) i jednocześnie na linii MTRX pojawi się logiczne zero, to silnik stacji wewnętrznej zostanie włączony. Dla tego napędu przerzutnik znajduje się na płycie głównej Amigi. Każda kolejna przyłączana stacja wymaga oddzielnego przerzutnika.

RDY
Po włączeniu silnika linia RDY (ReaDY) sygnalizuje Amidze osiągnięcie optymalnej prędkości obrotowej i gotowość do operacji zapis/odczyt. Jeśli silnik jest wyłączony linia ta jest używana do specjalnej identyfikacji (patrz dalej).

DRES
Linia DRES (Drive RESet) jest podłączona do linii RESET Amigi i jest wykorzystana do inicjalizacji przerzutnika (przerzutników) co powoduje wyłączenie silników wszystkich aktywnych w danej chwili napędów.

DKRD
Dane odczytywane z dysku w napędzie uaktywnionym przez SELx przekazywane są do Amigi linią DKRD (DisK Read Data). Podłączona jest ona do linii DKRD w układzie Paula.

DKWD
Jak wyżej, lecz dotyczy zapisu danych (DisK Write Data).

DKWE
Linia DKWE (DisK Write Enable) przełącza napęd z odczytu na zapis. Stan aktywny (niski) powoduje zapis na dyskietce.

SIDE
Linia SIDE wybiera stronę dysku na której ma się odbywać zapis/odczyt. Stan wysoki na tej linii powoduje wybranie strony (głowicy) o numerze 0 (dolna strona dyskietki).

WPRO
Linia WPRO (Write PROtect) sygnalizuje Amidze, że włożona do stacji dyskietka jest zabezpieczona przed zapisem. Stanem aktywnym tej linii jest stan niski.

STEP
Narastające zbocze sygnału na tej linii powoduje przesunięcie głowicy o jedną ścieżkę do przodu lub do tyłu w zależności od stanu na linii DIR.

DIR
Linia DIR (DIRection) wybiera kierunek przesuwu głowicy po impulsie STEP. Stan niski powoduje przesuwanie do środka dyskietki, wysoki zaś w kierunku obrzeża. Ścieżka 0 jest pierwszą ścieżką licząc od obrzeża.

TK0
Linia TK0 (TracK 0) jest ustawiana w stanie aktywnym niskim jeśli głowica wybranego napędu znajdzie się nad ścieżką zerową.

INDEX
Krótki pojedynczy impuls na tej linii jest wysyłany przez napęd w trakcie każdego obrotu dyskietki (pomiędzy początkiem i końcem ścieżki).

CHNG
Linia CHNG (CHaNGe) sygnalizuje Amidze zmianę dyskietki. Dopóki dysk znajduje się w napędzie linia ta ma stan wysoki. W momencie wyjęcia dyskietki linia ta przyjmuje stan niski i Amiga uaktywnia linię STEP (dobrze znane zjawisko „klikania") w regularnych odstępach czasu. Jest to konieczne do wykrycia momentu w którym do kieszeni stacji zostanie włożona nowa dyskietka. Po włożeniu dyskietki linia CHNG przyjmuje stan logicznej jedynki.

INUSE
Ta linia istnieje tylko na złączu zewnętrznych stacji. Jeśli Amiga ustawi tą linię w stan niski, napęd uaktywnia diodę LED sygnalizującą pracę stacji. Najczęściej linia ta jest podłączona do linii MTR i dioda zapala się w momencie gdy silnik stacji pracuje.

Do rozpoznania kiedy napęd jest podłączony do komputera używany jest specjalny tryb identyfikacji. Rozpoznanie przebiega w następujący sposób: linia MTR testowanej stacji jest włączana a następnie wyłączana. Powoduje to wyzerowanie szeregowego rejestru przesuwanego w napędzie. Poszczególne bity danych mogą być odczytywane przez uaktywnienie odpowiedniej linii SELx i odczytanie wartości na linii RDY jako bitu danej, po czym należy ustawić stan wysoki na linii SELx. Operacja taka musi być powtórzona 32 razy. Pierwszy odebrany bit jest najbardziej znaczący (MSB). Poniżej przedstawione są wartości odpowiadające różnym napędom:

$0000 0000 Napęd nie podłączony
$FFFF FFFF Standardowy napęd 3.5"/880 KB
$5555 5555 Napęd 5.25", 2x40 ścieżek

Wszystkie sygnały dostępne na złączu Shugart są aktywne w stanie niskim. Jest to spowodowane zastosowaniem w Amidze buforów z otwartym kolektorem (74LS07).

Cztery wejścia (CHNG, WPRO, TKO i RDY) są podłączone (w podanej kolejności) bezpośrednio do linii PA4-PA7 układu CIA-A. Wyjścia STEP, DIR, SIDE, SEL0, SEL1, SEL2, SEL3 i MTR są podłączone do linii PBO-PB7 układu CIA-B i poprzez bufory 74LS07 dalej do złącza stacji wewnętrznej i zewnętrznych. Pozostałe linie (DKRD, DKWD i DKWE) są podłączone do układu Paula. Z wyjątkiem linii MTRX i sygnałów SEL podłączenia do wewnętrznej i zewnętrznych stacji są takie same. Wewnętrzna stacja jest podłączona do linii SEL0, natomiast jej linia MTR do przerzutnika na płycie głównej.

Tyle teorii, teraz o podłączaniu napędów IBM do Amigi. Nadają się najlepiej do tego celu napędy 720 KB/3.5" oraz rzadko spotykane napędy 720 KB/5.25" (stosowane swego czasu m.in. w Elwro 800 Junior). Możliwe jest zastosowanie napędów 1.44 MB/3.5" oraz 1.2 MB/5.25" po przełączeniu ich w tryb 720 KB. Ponieważ przełączenia dokonuje się w różny sposób dla różnych napędów, odsyłam Czytelników do instrukcji danego napędu.

NIE NADAJĄ SIĘ DO NASZYCH ZASTOSOWAŃ NAPĘDY 360 KB/5.25" (chyba, że ktoś chce odczytywać dyski w formacie IBM/360 KB — format Amigi nie może być w tych stacjach odczytany). Z moich doświadczeń wynika, że najprościej podłączyć napędy firm Teac i Chinon (przy czym te pierwsze są lepszej jakości). Ponieważ napędy



tego typu są wyposażone w złącze Shugart kompatybilne z Amigą, nasze zadanie polega wyłącznie na zbudowaniu przerzutnika analogicznego do istniejącego na płycie głównej Amigi. Schemat takiego przerzutnika przedstawia rysunek 1.

Zwróć uwagę, że wykorzystujemy tylko połowę każdego z układów scalonych, można więc się pokusić na wykorzystanie drugiej połówki do wykonania drugiego przerzutnika. W ten sposób będziesz w stanie przyłączyć dwa napędy zewnętrzne zamiast jednego. Różnica będzie polegać tylko na podłączeniu wejść bramki N1 do linii SEL2 (napęd będzie widziany jako df2:).

W niektórych napędach wymagane jest dodatkowo odpowiednie ustawienie zworek (jumperów). Operacja ta jest prosta pod warunkiem, że Twój napęd był przy zakupie wyposażony w instrukcję obsługi w której podano dokładny opis wspominanych zworek. Brak takiego opisu to w większości wypadków ból głowy i mnóstwo wieczorów spędzonych na przestawianiu stada zworek.

W tym miejscu należy przestrzec wszystkich potencjalnych amatorów podłączania napędu do Amigi: błędy w montażu mogą spowodować uszkodzenie jednego lub obu portów CIA. Dlatego WSZELKICH PODŁĄCZEŃ NALEŻY DOKONYWAĆ DOPIERO PO SPRAWDZENIU POPRAWNOŚCI MONTAŻU. Jeśli nie czujesz się zbyt mocny w posługiwaniu się lutownicą i schematami prościej i chyba nawet TANIEJ będzie zakupić stację gotową.

JERZY DUDEK

B1,B2 - 74LS38
F1 - 74LS74
Zlacze SHUGART w stacji
Zlacze EXTERNAL DRIVE w Amidze
4 INUSE
16 MTR
2 CHNG
8 INDEX
10 SEL0
18 DIR
20 STEP
22 DKWD
24 DKWE
26 TK0
28 WPRO
30 DKRD
32 SIDE
34 RDY
1 +5V
2,3 masa
4 +12V
Z l a c z e zasilania w napedzie
4*1 kohm
+5V
D
CLK
PR
CLR
Q
F1
B1
B2
10
8
11
22
21
19
18
17
16
15
14
2
13
1
12
3,4,5,6,7
23

Podstawy asemblera: UKŁAD AGNUS i jego rejestry

Obok przedstawiam listę rejestrów układu „Agnus". Adresem bazowym dla obszaru rejestrów jest $dff000. Każdy z rejestrów zajmuje dwa bajty długości. Aby obliczyć adres absolutny interesującego nas rejestru, należy do adresu bazowego obszaru rejestrów dodać wartość z tabeli. Przykład: adresem rejestru COPCON jest $dff000 + $02e czyli $dff02e.

Zapraszam również do zapoznania się z artykułem z cyklu OD KUCHNI w którym J. Dudek omawia układ Agnus.

W ramce A przedstawiono listę rejestrów do których Copper ma dostęp tylko wtedy, gdy rejestr COPCON zawiera wartość 1 oraz (ramka B) których Copper ma dostęp zawsze.

BARTOSZ SMAGA

NAZWA	ADRES	FUNKCJA
COP1LCH	080	Bity 16–18 adresu pierwszej Copper listy
COP1LCL	082	Bity 1–15 adresu pierwszej Copper listy
COP2LCH	084	Bity 16–18 adresu drugiej Copper listy
COP2LCL	086	Bity 1–15 adresu pierwszej Copper listy
COPJMP1	088	Skok do początku pierwszej Copper listy
COPJMP2	08a	Skok do początku drugiej Copper listy
COPINS	08c	Rejestr rozkazów Coppera
DIWSTRT	08e	Górny lewy narożnik okna wizyjnego
DIWSTOP	090	Dolny prawy narożnik okna wizyjnego
DDFSTRT	092	Pozycja pozioma początku DMA płaszczyzny bitowej
DDFSTOP	094	Pozycja pozioma końca DMA płaszczyzny bitowej
DMACON	096	Zapis rejestru sterującego DMA
AUD0LCH	0a0	Bity 16–18 adresu danych dźwiękowych kanału 0
AUD0LCL	0a2	Bity 1–15 kanału 0
AUD1LCH	0b0	Bity 16–18 adresu danych dźwiękowych kanału 1
AUD1LCL	0b2	Bity 1–15 kanału 1
AUD2LCH	0c0	Bity 16–18 adresu danych dźwiękowych kanału 2
AUD2LCL	0c2	Bity 1–15 kanału 2
AUD3LCH	0d0	Bity 16–18 adresu danych dźwiękowych kanału 3
AUD3LCL	0d2	Bity 1–15 kanału 3
BPL1PTH	0e0	Bity 16–18 adresu płaszczyzny bitowej 1
BPL1PTL	0e2	Bity 1–15 adresu płaszczyzny bitowej 1
BPL2PTH	0e4	Bity 16–18 adresu płaszczyzny bitowej 2
BPL2PTL	0e6	Bity 1–15 adresu płaszczyzny bitowej 2
BPL3PTH	0e8	Bity 16–18 adresu płaszczyzny bitowej 3
BPL3PTL	0eA	Bity 1–15 adresu płaszczyzny bitowej 3
BPL4PTH	0eC	Bity 16–18 adresu płaszczyzny bitowej 4
BPL4PTL	0ee	Bity 1–15 adresu płaszczyzny bitowej 4
BPL5PTH	0f0	Bity 16–18 adresu płaszczyzny bitowej 5
BPL5PTL	0f2	Bity 1–15 adresu płaszczyzny bitowej 5
BPL6PTH	0f4	Bity 16–18 adresu płaszczyzny bitowej 6
BPL6PTL	0F6	Bity 1–15 adresu płaszczyzny bitowej 6
BPLCON0	100	Rejestr sterujący 0 płaszczyzny bitowej
BPL1MOD	108	Moduły płaszczyzny bitowej (płaszczyzny nieparzyste)
BPL2MOD	10a	Moduły płaszczyzny bitowej (płaszczyzny parzyste)
SPR0PTH	120	Bity 16–18 danych duszka 0
SPR0PTL	122	Bity 1–15 danych duszka 0
SPR1PTH	124	Bity 16–18 danych duszka 1
SPR1PTL	126	Bity 1–15 danych duszka 1
SPR2PTH	128	Bity 16–18 danych duszka 2
SPR2PTL	12a	Bity 1–15 danych duszka 2
SPR3PTH	12c	Bity 16–18 danych duszka 3
SPR3PTL	12e	Bity 1–15 danych duszka 3
SPR4PTH	130	Bity 16–18 danych duszka 4
SPR4PTL	132	Bity 1–15 danych duszka 4
SPR5PTH	134	Bity 16–18 danych duszka 5
SPR5PTL	136	Bity 1–15 danych duszka 5
SPR6PTH	138	Bity 16–18 danych duszka 6
SPR6PTL	13a	Bity 1–15 danych duszka 6
SPR7PTH	13c	Bity 16–18 danych duszka 7
SPR7PTL	13e	Bity 1–15 danych duszka 7
SPR0POS	140	Pozycja początkowa duszka 0 (X i Y)

SPR0CTL	142	Rejestr sterujący i pozycja końcowa duszka 0
SPR1POS	148	Pozycja początkowa duszka 1 (X i Y)
SPR1CTL	14a	Rejestr sterujący i pozycja końcowa duszka 1
SPR2POS	150	Pozycja początkowa duszka 2 (X i Y)
SPR2CTL	152	Rejestr sterujący i pozycja końcowa duszka 2
SPR3POS	158	Pozycja początkowa duszka 3 (X i Y)
SPR3CTL	15a	Rejestr sterujący i pozycja końcowa duszka 3
SPR4POS	160	Pozycja początkowa duszka 4 (X i Y)
SPR4CTL	162	Rejestr sterujący i pozycja końcowa duszka 4
SPR5POS	168	Pozycja początkowa duszka 5 (X i Y)
SPR5CTL	16a	Rejestr sterujący i pozycja końcowa duszka 5
SPR6POS	170	Pozycja początkowa duszka 6 (X i Y)
SPR6CTL	172	Rejestr sterujący i pozycja końcowa duszka 6
SPR7POS	178	Pozycja początkowa duszka 7 (X i Y)
SPR7CTL	17a	Rejestr sterujący i pozycja końcowa duszka 7

NAZWA	ADRES	FUNKCJA
BLTDDAT	000	Dane wyjściowe z blittera do RAM
DMACONR	002	Rejestr odczytu sterownika DMA
VPOSR	004	Starszy bajt MSB pozycji Y
VHPOSR	006	Pozycja X i Y wiązki wizyjnej
DSKPTH	020	Bity 16–18 adresu DMA dysku
DSKPTL	022	Bity 1–15 adresu DMA dysku
REFPTR	028	Licznik odświeżania
VPOSW	02A	Zapis MSB pozycji Y wiązki
VHPOSW	02C	Zapis pozycji X i Y wiązki
COPCON	02E	Rejestr sterujący Coppera

NAZWA	ADRES	FUNKCJA
BLTCON0	040	Rejestr sterujący nr 0 blittera
BLTCON1	042	Rejestr sterujący nr 1 blittera
BLTAFWM	044	Maska pierwszego słowa danych z A
BLTALWM	046	Maska ostatniego słowa danych z A
BLTCPTH	048	Bity 16–18 adresu danych źródłowych C
BLTCPTL	04a	Bity 1–15 adresu danych źródłowych C
BLTBPTH	04c	Bity 16–18 adresu danych źródłowych B
BLTBPTL	04e	Bity 1–15 adresu danych źródłowych B
BLTAPTH	050	Bity 16–18 adresu danych źródłowych A
BLTAPTL	052	Bity 1–15 adresu danych źródłowych A
BLTDPTH	054	Bity 16–18 adresu danych źródłowych D
BLTDPTL	056	Bity 1–15 adresu danych źródłowych D
BLTSIZE	058	Bit startu i rozmiar okna blittera
BLTCMOD	060	Moduł blittera dla danych źródłowych C
BLTBMOD	062	Moduł blittera dla danych źródłowych B
BLTAMOD	064	Moduł blittera dla danych źródłowych A
BLTDMOD	066	Moduł blittera dla danych źródłowych D
BLTCDAT	070	Rejestr C danych źródłowych blittera
BLTBDAT	072	Rejestr B danych źródłowych blittera
BLTADAT	074	Rejestr A danych źródłowych blittera



AMIGA dla początkujących

(część II)

CLI to jednak nie wszystko, jest jeszcze Workbench. Dostajemy się do niego za pomocą instrukcji „loadwb". Jeśli teraz zmniejszysz okno CLI za pomocą gadgetu w jego prawym dolnym rogu, zobaczysz ikonę dyskietki systemowej. Możesz teraz włożyć jakąkolwiek inną dyskietkę — po chwili pojawi się również i jej ikona.

Ikona może mieć dwa stany: nieaktywny i aktywny. Uaktywniamy ją po prostu „klikając" na niej lewym przyciskiem myszy — zwykle zmienia ona wtedy kolory, istnieje również możliwość zdefiniowania drugiego wzoru, ukazującego się po wywołaniu ikony. Dwukrotne kliknięcie zawsze wywołuje reakcję — jeśli jest to ikona katalogu albo dyskietki, wyświetlany jest jej katalog, jeśli ikona symbolizuje program, jest on wczytywany i uruchamiany.

Wywołaj więc katalog dyskietki systemowej. Na ekranie pojawi się otwarte okienko zawierające (lub nie) ikony programów i katalogów. Okno katalogu głównego ma zawsze z lewej strony wskaźnik zapełnienia — pionową skalę od litery E (empty, czyli pusty) do F (full, czyli pełny). Można w ten sposób niezbyt dokładnie, ale za to szybko zorientować się w ilości wolnego miejsca na dyskietce.

Przyjrzyj się teraz zawartości okna — widać w nim różne ikony: te w kształcie szuflad odpowiadają podkatalogom, ikona w kształcie kosza odpowiada katalogowi służącemu do usuwania plików, zaś ikona przypominająca okno wywołuje Shell (jest to odpowiednik CLI, pozwalający na wygodną edycję wprowadzanych poleceń jak również wywoływania instrukcji wydanych wcześniej).

W Amidze, w odróżnieniu od komputerów takich jak Macintosh czy Atari ST, system operacyjny nie tworzy ikon sam. Aby pojawiła się ikona np. dla pliku „ABC", konieczna jest obecność w tym samym katalogu drugiego pliku, definiującego ikonę (jego nazwa musi mieć rozszerzenie „.info". To samo dotyczy katalogów — weźmy dla przykładu katalog „Prefs" w głównym katalogu (czyli tym, z którego „Prefs" się „wywodzi") znajduje się plik „Prefs.info". Zastosowane przez programistów systemu operacyjnego rozwiązanie ma dużo zalet, ma też niestety i wady. Do zalet można zaliczyć fakt, że ikony mogą być kolorowe i urozmaicone, poza tym, nie ma na ekranie bałaganu, gdyż pojawiają się tylko ikony tych plików i katalogów, których chcemy — w przeciwnym wypadku pojawiły by się zupełnie zbędne ikony katalogów systemowych lub plików mniej ważnych.

Sporą wadą jest niestety konieczność ciągłego stosowania programów w rodzaju Disk Master, bowiem operacje na plikach i katalogach bez ikon są niedozwolone.

Warto jeszcze wspomnieć o bardzo ciekawej możliwości związanej z ikonami. Rozróżniamy ich kilka typów, powodujących „otwarcie" katalogu, uruchomienie programu, oraz automatyczne uruchomienie programu z parametrem. Dowcip polega na tym, że to właśnie w ikonie jest podawana nazwa programu do wykonania oraz parametry. W ten sposób, jednym kliknięciem możemy wywoływać wykonanie całych ciągów poleceń (np. parametrem jest plik tekstowy, a zdefiniowany program go po prostu wykonuje). Często edytory tekstu do plików tekstowych dogrywają swoje ikony — wtedy kliknięcie na ikonie tekstu powoduje automatyczne załadowanie edytora, i wczytanie żądanego dokumentu.

Wiemy już dużo o ikonach, teraz możemy zapoznać się z menu u góry ekranu. Pomysł został wzięty z komputera Macintosh, jednak ich wywoływanie jest rozwiązane lepiej. Listę menu wywołujemy wciśnięciem prawego przycisku myszy. Następnie, trzymając go, podjedź kursorem na samą górę. W momencie nałożenia wskaźnika na nazwę menu, rozwija się lista opcji. Wyboru można dokonać na dwa sposoby: albo ustawiamy kursor na interesującej nas opcji i puszczamy przycisk myszy, albo cały czas trzymając prawy przycisk, na każdej interesującej nas opcji klikamy lewym przyciskiem, a prawy zwalniamy dopiero na końcu. Ta druga technika jest bardzo przydatna przy wyborze wielu opcji naraz.

Przyjrzyjmy się więc opcjom. Pierwsza w menu „Workbench", „Open", robi dokładnie to samo co podwójne kliknięcie lewym przyciskiem myszy. Następna, „Close", zamyka okno głównego katalogu dysku. Opcja „Duplicate" służy do kopiowania dyskietek, jednak działa ona na tej zasadzie, że uruchamia systemowy program kopiujący, nadający się (niestety!) do wszystkiego poza kopiowaniem. Opcja „Rename" działa na szczęście samodzielnie — uaktywniamy ikonę dysku, katalogu lub pliku, i po wywołaniu „Rename" możemy wygodnie zmienić jego nazwę. Ciekawą opcją jest „Info" (uwaga: wymaga obecności na dysku systemowym biblioteki „info.library"). Opcja ta otwiera okno zawierające informacje o ikonie — nie wydaje się to dużo, jednak można przy jej pomocy pozmieniać praktycznie wszystko — możliwość zapisu lub kasowania pliku (w ramce „STATUS"), wielkość stosu dla programu („STACK"), ewentualne dodatkowe parametry („TOOL TYPES") oraz program wykonawczy („DEFAULT TOOL"). Ostatnia opcja w menu „Workbench" — „Discard" usuwa po prostu plik lub katalog, którego ikonę uaktywniliśmy.

Następne menu to „Disk" — zawiera tylko dwie opcje, prawie nigdy zresztą nieużywane. Pierwsza z nich to „Empty Trash" — powoduje usunięcie wszystkich plików z katalogu „Trashcan". Druga, „Initialize", formatuje dyskietki, jednak, podobnie jak „Duplicate", korzysta z programów zawartych na dysku systemowym.

Ostatnie menu, „Special", zawiera opcje o zastosowaniu ogólnym. „Clean Up" ma za zadanie ułożenie ikon w ich oknie — należy uaktywnić ikonę dysku lub katalogu, którego okno jest już otwarte, po czym wybrać opcję „Clean Up". „Last Error" powoduje ponowne wyświetlenie ostatniego komunikatu (Workbench Screen"). Opcja „Redraw" każe systemowi operacyjnemu narysować ponownie cały ekran — jest ona przydatna bardzo rzadko, gdyż system operacyjny bardzo rzadko „zapomina" o uaktualnieniu zawartości ekranu. Dość ciekawa jest opcja „Snapshot". Powoduje ona zapisanie na dyskietce położenia ikon lub okien. Jeśli na przykład chcesz, aby ikona dysku pojawiła się na środku ekranu, przesuwamy ją tam (przytrzymując na niej naciśnięty lewy przycisk myszy), po czym wybieramy opcję „Snapshot". Podobnie można postępować z ikonami plików i katalogów, a także z oknami katalogów (należy uaktywnić ikonę katalogu, potem jego okno, i dopiero teraz wybrać opcje „Snapshot").

Została nam już tylko opcja „Version" — podobnie jak „Info", opcja ta korzysta z biblioteki (version.library). Jedynym jej zadaniem jest wyświetlenie informacji o wersji systemu operacyjnego Kickstart i oprogramowania (Workbench)

Na koniec warto jeszcze wspomnieć o kilku sztuczkach związanych właśnie z Workbenchem. Pierwsza z nich to użycie klawisza SHIFT — jeśli przytrzymamy go, to możliwe jest uaktywnienie kilku ikon na raz. Daje to dwie korzyści: możemy wykonywać operacje na wielu plikach tylko raz wywołując opcje z menu, mamy też możliwość „ręcznego" podania parametru — najpierw uaktywniamy ikonę pliku, który będzie parametrem (na przykład obrazka do wyświetlania), a potem trzymając SHIFT uruchamiamy dwukrotnym kliknięciem program wykonawczy. Drugi sposobik powinien okazać się bardzo przydatny dla amigowców bez rozszerzeń pamięci; jak wiadomo, za pomocą ogólnie dostępnych programów mogą oni bowiem kopiować tylko mniejsze pliki. Wystarczy teraz tylko dograć do takiego pliku dowolną ikonę i skopiować go przy pomocy Workbencha. Należy otworzyć okna dysków źródłowego i docelowego, po czym przesunąć ikonę interesującego nas pliku z okna źródłowego do docelowego. W ten sposób można kopiować nawet najdłuższe pliki.

ANDRZEJ BOBEK

BIRDS OF PREY

Na ten symulator ostrzyłem sobie zęby dość długo; pierwsze słuchy na jego temat zaliczały go do grona „niezwykłych”. Kilka tygodni później długo oczekiwane dyskietki znalazły się wreszcie w redakcyjnej Amidze.

Pierwsze wrażenia są rzeczywiście niezwykłe. Znakomicie przygotowane i bardzo efektowne intro (zajmujące całą pierwszą dyskietkę) pokazuje co naprawdę potrafi Amiga.

Birds of Prey dają Ci do wyboru 40 typów samolotów: od dawno już nieużywanego F-104 Starfighter do Boeinga 747-400. Lista jest rzeczywiście imponująca i znajdują się na niej samoloty doświadczalne (X-15), Antonow 26, C-130 Hercules, F-117, F-14, Mirage i wiele innych. W chwili wyboru na ekranie pojawia się sylwetka „drapieżnego ptaka” i jego dane techniczne wraz z króciutką charakterystyką. Całość zrobiona jest bardzo efektownie.

Aby pilotowi się nie nudziło ma on do wyboru 12 różnych misji od przechwycenia począwszy a na bombardowaniu dalekiego zasięgu skończywszy. Po uzbrojeniu samolotu komputer przenosi Cię do kabiny. Zapowiadało się ciekawie: pierwsze zadanie polegało na uruchomieniu silników i wykołowaniu z hangaru. Operacja ta jest (po kilku próbach) stosunkowo prosta, gorzej jest z samym kołowaniem i trafieniem w próg drogi startowej. Zjechanie z pasa w czasie startu powoduje gwałtowny spadek prędkości i duże kłopoty ze wzbiciem się w powietrze.

Po pierwszej misji, rozsmakowałem się w lataniu i postanowiłem zostać kapitanem Boeinga 747. Tu spotkało mnie pierwsze rozczarowanie: na ekranie pojawił się wystrój kabiny znany mi z poprzedniej misji. Niestety twórcy nie zadali sobie trudu, aby odwzorować dokładnie urządzenia pokładowe i wnętrza kabin wszystkich 40 typów samolotów. Takie rozwiązanie ma jednak pewną zaletę: użytkownik nie musi raz po raz sięgać po instrukcję obsługi, aby dowiedzieć się co wskazuje określone urządzenie.

Dużą niespodziankę (po „oblatywaniu” F-15 i A-10 w wersjach dla Amigi) sprawiły mi stery — są one nareszcie wrażliwe, a na przeniesienie ruchu nie trzeba czekać wieczności. Operacje startu i lądowania nie są zbyt trudne i można się ich nauczyć po kilku lotach próbnych. Tu odkryłem małe niedopracowanie: C-130 Hercules, samolot transportowy przystosowany do lądowisk każdego typu nie „umie” wystartować z trawy.

Na uwagę zasługuje również instrukcja obsługi w której jednak nie ustrzeżono się (przynajmniej w pierwszych egzemplarzach) gafy: pytania-hasła zabezpieczające program są na ekranie zadawane po angielsku, natomiast odpowiedzi na nie przetłumaczono w instrukcji na język polski. Poza tą gafą podręcznik jest w zasadzie bez zarzutu.

Ogólnie rzecz biorąc „Birds of Prey” są na pewno programem bardzo interesującym, tym niemniej, po głębszym poznaniu, pozostaje jednak pewien niedosyt. W moim wypadku spodziewałem się chyba większej różnorodności, wiernego odwzorowania przyrządów pokładowych, nieco bardziej wysublimowanej akcji. Nie oznacza to jednak, że nie można z tym programem spędzić kilku czy nawet kilkunastu ciekawych godzin, czego życzę wszystkim Czytelnikom „C&A”.

SIDEWINDER

DYSTRYBUTOR:
IPS Computer Group, Warszawa, ul. Okrężna 3, tel. 642-2766.

* Program : BIRDS OF PREY
* Rodzaj gry : symulator lotu
* Producent : **Electronic Arts**
* Komputer : **Amiga**
* Wymagania : -

POPULOUS II

„Pociąg Zeusa do śmiertelnych kobiet był legendarny, ale nigdy nie sprawiało mu przyjemności przychodzenie do nich w postaci BOGA. Zamiast tego przybierał postać łabędzia, fauna lub innej istoty. Takie właśnie było moje poczęcie ... Zawsze wiedziałem, że jestem niezwykłym dzieckiem, ale także wiedziałem, że Zeus nigdy mnie nie rozpozna — nigdy nie pozwoli zasiąść na panteonie ... chyba że, przeżyję zmagania olimpijskich bogów!”

Tu rozpoczyna się Twoja przygoda i gra Populous 2. Twoim zadaniem jest pokonanie 35-ciu bogów w zamian za co Zeus pozwoli ci zasiąść na Olimpie. Pierwsze co należy uczynić to stworzyć boską osobowość. Możesz ustalić swój wygląd i umiejętności. Ma to znaczący wpływ na grę, gdyż w zależności od wyglądu przeciwnicy będą różnie reagować na Twoje wyczyny. Umiejętności z kolei, mają wpływ na siłę i zasięg czarów jakie możesz rzucać.

Teraz można przystąpić do gry. Są dwa rodzaje gier: Conquest i Custom. W tym pierwszym (Konkwista) gra się po kolei we wszystkie „światy” ze wzrastającym poziomem trudności. W miarę gry, gdy wykazujesz się coraz większymi postępami, zdobywasz doświadczenie i umiejętności, Twój przeciwnik staje się silniejszy, warunki robią się ostrzejsze i zmieniają się dostępne efekty — czary.

W grze typu Custom, walczysz przy użyciu wszystkich czarów jakie zdobyłeś wcześniej grając w gry Conquest. W tym wariancie możemy grać z komputerem, lub z kolegą (niestety, gra dwuosobowa jest możliwa po sprzęgnięciu komputerów modemem lub portami szeregowymi. Możliwy jest także wybór opcji, w której komputer gra sam z sobą, Ty natomiast obserwujesz tę walkę.

Akcja Populous 2 toczy się w epoce helleńskiej. W czasie gry możemy spotkać hydrę, syreny, olbrzymy oraz wiele innych mitologicznych stworzeń. Zarówno grafika jak i czary jakie mogą rzucać bogowie są w stosunku do Populous 1 o wiele bardziej interesujące. Do dyspozycji mamy trzęsienia ziemi, wielkie fale powodziowe zatapiające ląd, wulkany, trąby powietrzne. Możesz również zsyłać plagi i także wiele innych podobnych kataklizmów. Kataklizmy zsyła się na wyznawców przeciwnika; własnymi wiernymi należy się zaś troskliwie opiekować i wyrównywać im ziemię, aby mogli się w zdrowiu rozmnażać. Pamiętaj: im więcej wyznawców ma Bóg, tym większa jest jego siła.



Już okładka gry Populous 2 przyciąga nasze oczy, lecz jest to tylko przedsmak wspaniałej grafiki jaką zobaczymy po uruchomieniu programu. Starannie opracowane szczegóły oraz wspaniałe efekty dźwiękowe to najmocniejsze atuty gry. Populous 2 ma też pewne walory edukacyjne. Otóż przed walką, kiedy oglądamy przeciwnika, ukazuje się krótka charakterystyka konkurencyjnego boga. Jest to znakomite przypomnienie mitologii.

Gra jest bardzo interesująca i wciągająca (grało mi się z wielką przyjemnością). Moim zdaniem jedyną wadą gry jest nienajlepsza muzyka — bardzo monotonna i zagłuszająca efekty. Ponadto muzyki raz włączonej nie można już wyłączyć. Do plusów zaliczam bardzo starannie wydaną instrukcję obsługi, która bardzo przystępnie wyjaśnia reguły gry. W pierwszej fazie dystrybutor rozpowszechniał grę z instrukcją anglojęzyczną, obecnie dostępna jest już wersja polska tego podręcznika. Dyskietka (oprócz programu) zawiera również specjalny plik pozwalający zainstalować grę na dysku twardym. Reasumując: dobra robota, dobry program, bardzo dobra zabawa.

JETBOY

Dystrybutor:
IPS Computer Group, Warszawa, tel. 642-2766.

* Rodzaj gry: role-playing
* Producent: **Electronic Arts**
* Komputer: **Amiga 500/2000/3000**
* Wymagania: 1 MB RAM

UKŁAD AGNUS

Agnus jest w Amidze najbardziej złożonym układem specjalizowanym — zawiera całą strukturę układów DMA. Każde z sześciu źródeł DMA ma swoją własną logikę sterującą. Wszystkie są podłączone do generatora adresu CHIP-RAM oraz do generatora adresu rejestru. Generatory te wytwarzają adres pożądanej komórki w obszarze CHIP-RAM i adres rejestru docelowego. W ten sposób logika układu DMA zapisuje dane pobrane z pamięci RAM do wybranego rejestru lub wpisuje zawartość rejestru do wybranej komórki pamięci RAM.

Do generatora adresu CHIP-RAM jest także podłączony licznik, generujący sygnały odświeżające pamięć RAM. Jest to niezbędne przy zastosowanych w Amidze dynamicznych pamięciach RAM.

Agnus zawiera również liczniki rastra i kolumn. Liczniki te generują także sygnały synchronizacji poziomej i pionowej. Sygnał synchronizacji poziomej sygnalizuje start nowej linii, natomiast synchronizacji pionowej — start nowej ramki. Sygnały synchronizacji mogą być doprowadzone z zewnątrz, co umożliwia zsynchronizowanie obrazu Amigi np. z obrazem z magnetowidu (Genlock).

Kolejnymi ważnymi elementami w Agnus są blitter i copper. Blitter jest specjalnym układem mogącym bardzo szybko przesuwać obszary pamięci. Zwalnia to procesor od wykonywania bardzo czasochłonnych operacji, zresztą blitter wykonuje je znacznie szybciej niż potrafiłby to zrobić procesor. Blitter może także wypełniać obszary pamięci oraz wykonywać na nich pewne operacje logiczne.

Copper jest jednym z koprocesorów. Jego program nazywany copper-listą zawiera tylko trzy instrukcje. Może on zmieniać zawartość różnych rejestrów układów specjalizowanych w momencie, gdy wiązka elektronów na ekranie osiągnie z góry ustaloną pozycję.

A oto poszczególne linie sygnałowe układu Agnus:

Szyna danych D0-D15:
Szesnaście linii danych przyłączonych bezpośrednio do szyny danych CHIP-RAM. Wewnątrz wszystkie rejestry są dołączone poprzez bufory do tej szyny.

Szyna adresowa rejestrów RGA0-RGA8:
Szyna ta jest dwukierunkowa. W czasie operacji DMA generator adresu rejestru umieszcza adres żądanego rejestru na tej szynie. Jeśli natomiast procesor potrzebuje dostępu do któregoś z rejestrów, linia ta jest ustawiona jako wejście i adres rejestru wybranego przez procesor jest przesyłany do dekodera adresu rejestru wewnątrz układu Agnus. Jeśli na szynie adresowej rejestrów występuje wartość 255 (SFF) to żaden rejestr nie jest wybrany.

Linie adresowe dla pamięci RAM DRA0--DRA8:
Linie te są podłączone do szyny adresowej CHIP-RAM. Są one jednokierunkowe (tylko wyjście) i uaktywniane przez Agnus, gdy konieczna jest operacja DMA. Linie te są podłączone za pośrednictwem multipleksera do szyny adresowej pamięci. Dzięki multiplekserowi mogą być one podłączone bezpośrednio do linii adresowych kości pamięci DRAM o pojemności 32 KB (oznaczenie 41256). Niestety, multiplekser występuje tylko w Amigach 500 i 2000; w starej Amidze 1000 RAM ma on tylko osiem linii adresowych. Dziewiąty bit DRA jest multipleksowany i używany do wyboru (przełączania) banków pamięci.

Linie zegarowe CCK i CCKQ:
Częstotliwość obu sygnałów zegarowych na tych liniach wynosi 3.58 MHz; jest to połowa częstotliwości zegara dla procesora. Sygnał CCKQ jest opóźniony w fazie względem sygnału CCK o ćwierć taktu zegara (90 stopni).

Linie kontroli szyn BLS, ARW i DBR:
Te linie są podłączone do logiki sterującej Amigi. Linia DBR (Data Bus Request) informuje logikę sterującą, że Agnus żąda dostępu do szyny danych w następnym cyklu. Linia ta ma wyższy priorytet niż żądanie dostępu do procesora. Jeśli Agnus wykonuje operacje zajmujące kilka cykli, procesor musi czekać, co tłumaczy powolność wykonywania programu umieszczonego w CHIP RAM.
Linia ARW (Agnus RAM Write) sygnalizuje logice sterującej chęć zapisu przez Agnus do pamięci RAM.
Linia BLS (BLitter Slow down) sygnalizuje Agnusowi, że procesor odczekał już trzy cykle dostępu do szyny danych. W zależności od stanów wewnętrznych układu, jeżeli jest to możliwe, Agnus jest odłączany od szyny na jeden cykl, aby mógł z niej skorzystać procesor.

Sygnały sterujące RES, INT3, DMAL:
Sygnał RES (RESet) jest podłączony bezpośrednio do linii RESET procesora.
Linia INTR3 (INTRerrupt at level 3) jest wyjściem i jest podłączona bezpośrednio do linii w układzie Paula o tej samej nazwie. Agnus używa tej linii do poinformowania układu logiki przerwań w Pauli, że któryś z jej elementów wytworzył przerwanie.
Linia DMAL (DMA request Line) jest również podłączona do Pauli, lecz tym razem w odwrotnym kierunku. Paula używa tej linii do poinformowania Agnusa o konieczności wykonania przesłania DMA.

Linie HSY, VSY, CSY i LP:
Linie HSY i VSY są normalnymi sygnałami synchronizacji poziomej (Horizontal SYnchronization) i pionowej (Vertical SYnchronization) i są używane do synchronizacji obrazu na monitorze. Sygnał CSY (Composite SYnchronization) jest sumą sygnałów HSY i VSY. Linie te mogą działać również jako wejścia, co — jak zaznaczono wcześniej — umożliwia synchronizację obrazu z zewnętrznym źródłem. Linia LP jest wejściem i umożliwia podłączenie pióra świetlnego (Light Pen).

FAT AGNUS

Wszystko, co napisano powyżej dotyczy wszystkich Amig. Jednak w nowszych jej typach wprowadzono udoskonaloną wersję układu Agnus, nazywaną Fat Agnus. Struktura pozostała nie zmieniona, we wnętrzu nowej kości umieszczono tylko wiele układów pozostających dotąd poza obudową Agnus. Wyeliminowano w ten sposób sporą liczbę prostych układów scalonych i innych elementów. Miało to na celu uproszczenie konstrukcji i w efekcie obniżenie ceny komputera. Różnice są następujące:

1. Cały system generacji sygnałów zegarowych jest umieszczony w Fat Agnus. Potrzeba tylko jednego zewnętrznego sygnału zegara o częstotliwości 28 MHz. Dzielenie tej częstotliwości dla pozostałych układów jest wykonywane obecnie wewnątrz Agnus. Sygnały tego bloku noszą nazwy: 28MHz, XCLK, XCLKEN, 7MHz, CCK, CCKQ i CDAC.

2. Bufor szyny adresowej został umieszczony wewnątrz układu Agnus, co umożliwia przyłączenie linii adresowych procesora bezpośrednio do linii A1-A18 Fat Agnusa. Dekoder adresów wykorzystuje teraz dodatkowo sygnały RAMEN (RAM ENable) i RGEN (ReGister ENable) do sygnalizacji, kiedy procesor potrzebuje użyć obszaru rejestrów lub CHIP--RAM. Oprócz tego Agnus jest połączony z procesorem za pomocą trzech sygnałów: UDS, LDS i PR/W (Processor Read/Write).

3. Oprócz powyższych zmian Fat Agnus zarządza teraz całkowicie pamięcią CHIP-RAM. Wytwarza on sygnały RAS i CAS wraz ze zmultipleksowanymi sygnałami adresowymi (RAS — Row Address Select, CAS — Column Address Select). Dodatkowo wprowadzono sygnały RAS1 i CAS1 do zarządzania rozszerzeniem pamięci (RAM Expansion).

Razem z Fat Agnus pojawił się w Amidze czwarty układ specjalizowany nazwany Gary. Przejął on funkcje dekodera adresów i kontrolera szyn. Wytwarza także sygnały VPA i DTACK dla procesora. Wewnątrz Gary'ego znajduje się przerzutnik dla wewnętrznej stacji dysków.

FATTER AGNUS

W nowszych modelach Amigi montowany jest układ Fatter Agnus różniący się od Fat Agnus tym, że moze zarządzać 1 MB pamięci CHIP--RAM (Fat Agnus tylko 0.5 MB).

A teraz „co i gdzie jest zamontowane?" Układ Fatter Agnus jest montowany w modelach Amigi 500 i 2000 z Kickstartem 1.3 i wyżej. Amiga z Kickstartem 1.2 zawiera układ Fat Agnus. Natomiast w Amidze 1000 znajduje się zwykły Agnus.

Opracował: JERZY DUDEK

Literatura: „Amiga System Programming Guide"

ROZSZERZENIE PAMIĘCI

AMI 2.5 MB dla AMIGI 500

Dla nikogo nie jest tajemnicą, że Amiga z 1 MB pamięci RAM to nadal za mało. Nie każdy jednak zdaje sobie sprawę, że różne rozszerzenia oferują różne rodzaje pamięci; zastosowanie nieodpowiedniego nie uchroni użytkownika przed komunikatem „NOT ENOUGH MEMORY". Przyczyna jest prozaiczna: wolna pamięć to FAST lub SLOW, a do niej nie mają dostępu procesory Amigi.

PRZEZNACZENIE, ZESTAW I GWARANCJA

Opisywane tu rozszerzenie powiększa pamięć CHIP-RAM z 512 KB do 1 MB, natomiast pamięć SLOW — do 1.5 MB. Urządzenie działa poprawnie TYLKO w Amigach 500 z systemem V1.3 wyposażonych w tzw. NOWĄ PŁYTĘ, z układem FATTER AGNUS potrafiącym obsługiwać 1 MB CHIP-RAM.

Atrakcyjność oferty podnosi możliwość zwrotu zakupionego wcześniej w firmie MIKROSERWIS rozszerzenia 512 KB w ramach opłaty za rozszerzenie 2.5 MB.

W skład zestawu wchodzi płytka z rozszerzeniem, podstawka pod kość GARY, wyłącznik oraz instrukcja obsługi. Producent udziela dwunastomiesięcznej gwarancji na swój wyrób na ogólnie przyjętych zasadach.

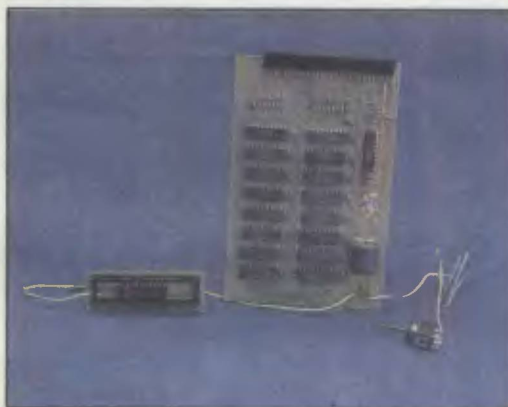
KONSTRUKCJA

Rozszerzenie jest wykonane na laminacie dwustronnym. Kości RAM są wlutowane bezpośrednio w płytkę (bez podstawek), co utrudnia ich wymianę w razie uszkodzenia. Aby zwiększyć niezawodność urządzenia zastosowano w nim wysokiej jakości firmy Siemens o czasie dostępu mniejszym niż 100 ns. Zegar zasilany jest z akumulatorka o pojemności 110 mAh. Wyłącznik pozwala na odłączanie rozszerzenia w razie potrzeby. Dodatkowe 0.5 MB pamięci CHIP-RAM zajmuje obszar adresowy od $80000 do $fffff, natomiast 1.5 MB SLOW-RAM — od adresu $c00000 do $d7ffff. Istnieje możliwość zakupu rozszerzenia w wersji bez zegara (zmniejsza to cenę o około 200 tys. zł.).

DOKUMENTACJA I INSTALACJA

Do rozszerzenia dołączona jest krótka instrukcja, wyjaśniająca jasno i przejrzyście punkt po punkcie sposób instalacji. Mniej zręcznym użytkownikom producent oferuje bezpłatną instalację przy zakupie (po uprzednim umówieniu się).

Aby zainstalować rozszerzenie należy otworzyć komputer, co może wiązać się z zerwaniem plomby gwarancyjnej. Po wsunięciu rozszerzenia w slot pod klawiaturą należy wyjąć z podstawki układ GARY, włożyć w jego miejsce płytkę dostarczaną z rozszerzeniem i wcisnąć w zainstalowaną na niej podstawkę wyjęty układ GARY. Następnie trzeba przeciąć wskazaną zworkę (jumper) na płycie komputera i przylutować w to miejsce kabelek wyprowadzony z płytki rozszerzenia. Zalecam stosowanie lutownic z jak najcieńszym grotem — kolby o mocy większej niż 15W mogą przepalić ścieżki.



W AKCJI

W praktyce rozszerzenie wyróżnia się głównie tym, że mamy do dyspozycji 2.5 MB Użytkownik ma do dyspozycji 1 MB CHIP-RAM oraz 1.5 MB SLOW-RAM. W rozszerzeniach dostępnych dotychczas podział ten był inny: 1.8 SLOW-RAM, ale za to CHIP-RAM pozostawała na poziomie 512 KB. Zniknęły również dość powszechne dotychczas przy większych rozszerzeniach kłopoty z zegarem: przez cały okres testowania zegar nie przestawił się samoczynnie ani razu i działał bardzo dokładnie.

WSPÓŁPRACA Z OPROGRAMOWANIEM

Ponieważ pamięć jest podzielona zgodnie z podziałem adresowym przewidzianym przez konstruktorów praca rozszerzenia nie różni się od pracy innych urządzeń tego typu. Problemy występują jedynie (Z WINY SYSTEMU OPERACYJNEGO) przy programach rezydentnych, pozostających w pamięci i działających także po inicjalizacji (reset) Amigi.

Po każdym wciśnięciu klawiszy CTRL-AMIGA-AMIGA kasowana jest CAŁA pamięć dzięki czemu niszczone są WSZYSTKIE zawarte w RAM programy. Nie jest to wynikiem złej konstrukcji rozszerzenia czy jego uszkodzenia, lecz konsekwencją działania systemu operacyjnego. Nie będą więc działać programy rezydentne, w tym bardzo wiele wirusów; z pamięci usuwane są jednak i pożyteczne programy, np. program obsługi emulatora A-Tonce czy rezydentny RAM-dysk (RAD:). Na RAM-dysk mam już lekarstwo — wystarczy skorzystać z programu SetPatch z katalogu "C". Zawiera on opcję uniemożliwiającą skasowanie RAD: podczas inicjalizacji.

WERDYKT

Omawiane rozszerzenie pamięci umożliwia rozbudowę obu typów pamięci jednocześnie i wydaje się być bardzo dobrą propozycją dla użytkowników, zamierzających wykorzystać swój komputer dla poważniejszych celów. Stosunkowo niska cena, wysoka jakość zastosowanych elementów i niezbyt skomplikowany montaż pozwalają mi uznać to rozszerzenie za bardzo pożyteczne rozwiązanie eliminujące problemy z pamięcią RAM na dłuższą metę.

ANDRZEJ BOBEK

PRODUCENT I DYSTRYBUTOR:
Firma „MIKROSERWIS", 80-288 Gdańsk, ul. Marusarzówny 6, telefon, (058)-48-50-63.

CENA: *2.0 mln złotych (kwiecień 1992)*

ZAMÓWIENIA:

Rozszerzenie można nabyć bezpośrednio w zakładzie, jak również wysyłkowo za pobraniem pocztowym, po uprzednim zamówieniu. Do tego doliczane są koszty przesyłki.

ZALETY:

— niespotykane i bardzo pożyteczne rozszerzenie CHIP-RAM
— zastosowanie wysokogatunkowych układów pamięci
— stabilny zegar
— niska cena

WADY:

— dość toporny wyłącznik
— kłopotliwa instalacja (z powodu konstrukcji Amigi)
— konieczność otwierania komputera

ACTION REPLAY
MK III dla AMIGI

Od razu na początku chcę zaznaczyć, że nie będę się koncentrował na różnicach z wersją Mk II z dwóch powodów: po pierwsze dlatego, że nie miałem dużej styczności z poprzednią wersją, po drugie potencjalni nabywcy nowych kart wcale niekoniecznie muszą być posiadaczami wersji MK II. Dodam jeszcze, że test AR MK II był opublikowany w C&A 02/92.

CO TO JEST

Amiga Action Replay Mk III jest to moduł (karta) przeznaczona dla niemal wszystkich komputerów Amiga, a więc A500 z pochodnymi, A2000/3000 (specjalna wersja) oraz A1000 (również wersja specjalna). Chwilowo nie ma jeszcze wersji dla Amigi 300 i 600, a to z powodu stosunkowo młodego wieku tego komputera. Model jaki otrzymaliśmy do testu był przeznaczony do Amigi 500 i 500 PLUS, lecz dystrybutor. firma Inter-Comp oferuje wszystkie modele.

KONSTRUKCJA

AR3, podobnie jak jego poprzednik jest wyposażony w przyciski FREEZE I RESET oraz w specjalny potencjometr spowalniający pracę komputera. Warto wspomnieć, że powtórzona została wada z poprzedniego modelu — karta nie jest „przelotowa", tzn. niemożliwe jest na przykład podłączenie AR3 i dysku twardego. Główną kość wykonano specjalną techniką montażu powierzchniowego co chroni ten moduł przed rączkami polskich „specjalistów" od podrabiania takich urządzeń. Moduł zawiera pamięć ROM o pojemności 128 KB.

GWARANCJA I SERWIS

Dystrybutor udziela dwunastomiesięcznej gwarancji na zakupione moduły. Ponieważ moduł nie ma części naprawialnych „naprawa" polega na wymianie całego modułu.



INSTRUKCJA OBSŁUGI

Pierwszym dużym plusem instrukcji jest fakt, iż napisano ją w języku angielskim. Polecenia zostały podzielone na kilka grup tematycznych, np. „trainer commands", „disk comands", „monitor commands" itp. Każde polecenie jest opisane w instrukcji dość szczegółowo, podano również przykłady jak należy z takich poleceń korzystać. Dodatkowo użytkownik ma w każdej chwili dostępną pomoc w postaci kilku ekranów wypełnionych krótkim opisem każdej instrukcji. W zasadzie nic dodać, nic ująć.

INSTALACJA

Proces instalacji jest bardzo prosty i wykonać go może każdy — należy wyłączyć komputer, zdjąć zaślepkę z lewej strony komputera i wsunąć (bez użycia przesadnej siły) moduł. Jeżeli wszystko zostało wykonane poprawnie (po uruchomieniu powinna świecić się dioda POWER w AR3) naszym oczom ukaże się gustowny napis ACTION REPLAY III i nazwiska twórców. Po mniej więcej dwóch sekundach pojawi się znajoma „łapa z dyskietką" (animacja wkładanej dyskietki w Amidze 500+...) i odtąd AR3 jest niewidoczny dla systemu czekając tylko na wciśnięcie przycisku FREEZE. Można to uczynić w dowolnej chwili bez zniszczenia żadnego z parametrów systemu.

W AKCJI

AR3 pozwala na zatrzymania pracy komputera w dowolnym momencie bez kasowania zawartości pamięci i rejestrów procesorów Amigi. Po „zamrożeniu" pamięci użytkownik może wykonać szereg różnorodnych czynności na „zamrożonym" programie, dyskietkach czy innych urządzeniach zewnętrznych. Na początek wymienię tylko ważniejsze opcje dostępne dzięki AR3. Moduł ten ma:

a) wbudowany kompletny monitor języka maszynowego z możliwością śledzenia wykonywania programu,
b) „ripper" czyli program pozwalający na wyszukiwanie i „wyciąganie" z programów grafiki i muzyki (pod warunkiem, że napisana została na jednym z programów typu „tracker", np. ProTracker) oraz dźwięków (sampli),
c) „trainer" czyli opcję półautomatycznego wyszukiwania „nieśmiertelności" w dowolnej grze (pomocne dla graczy),
d) tzw. „Burst Nibbler" czyli program kopiujący podobny do X-Copy ale o większych możliwościach. Burst Nibbler jest w stanie kopiować dyskietki np. w formacie IBM//PC czy Atari ST.

Posiadacz AR3 ma pełną kontrolę nad maszyną, włączając w to możliwość konfigurowania pojemności pamięci, dostępnych urządzeń zewnętrznych (np. stacje dysków), czy skuteczną obronę przed KAŻDYM wirusem. Jak wiadomo w Polsce najgroźniejszym wirusem jest Saddam — AR3 stanowi dla niego granicę nie do przeskoczenia. Jakby nie było jeszcze tego mało projektanci układu stworzyli możliwość teoretycznie płynnego regulowania szybkości komputera od 100 do ok. 20%.

Moduł jest aktywizowany wciśnięciem przycisku FREEZE. Po mniej więcej sekundzie ukazuje się ekran tekstowy z napisem „powitalnym". Od tego momentu możesz w pełni korzystać z oprogramowania zawartego w AR3. Rzut oka na zawartość instrukcji i już można powiedzieć, że nie wyrzuciliśmy pieniędzy w błoto — można zrobić (prawie)

wszystko czego dusza zapragnie. Nie będę tu przytaczał instrukcji obsługi; skupię się raczej na wskazaniu Czytelnikom celów do jakich mogą ten moduł wykorzystać.

GRACZE
— wyszukiwanie nieśmiertelności w grach (omówione na przykładach w instrukcji obsługi)
— program kopiujący dostępny w każdej chwili
— sprzętowy autofire

UŻYTKOWNICY
— „wyciąganie" dowolnego obrazka, dźwięku i muzyczek
— pełna informacja o systemie,
— możliwość wykorzystania pamięci jako RAM-dysk (sterowanego przez AR3; taki RAM-dysk jest w stanie „przeżyć RESET
— kodowanie dysków i dyskietek),
— obrona przed wirusami.

PROGRAMIŚCI
— kompletny system śledzenia programu,
— „no more guru" czyli przejęcie błędnych programów

TORTURY

Testy przeprowadzałem na Amidze 500 z 1.5 MB pamięci co pozwoliło na wykorzystanie wszystkich możliwości AR3 (na Amigach z 512 KB niedostępny jest np. ram-dysk). Żaden program nie zauważył obecności modułu, ponadto nie wystąpiły problemy z przywróceniem do życia „zamrożonych" programów.

Testując urządzenie na grach w każdej próbowałem uzyskać nieśmiertelność. Niestety, niektóre gry nie poddały się tak łatwo (np. Super Hang-On) i użytkownik nie znający języka maszynowego nie uzyska nieśmiertelności. Dla kontrastu mogę powiedzieć, że gra Badlands poddała się w pierwszym podejściu, tak jak Hollywood Poker Pro (ucieszy to zapewne użytkowników tego programu). Co do „wycinania" grafiki, to jest to możliwe w tych programach w których nie wykorzystano specjalnych technik tworzenia grafiki, np. generujących z jednego koloru sto barw — w takich wypadkach AR3 nie pomoże.

Program kopiujący Burst Nibbler jest to specjalna „replayowa" wersja podobnego, wydanego kilka lat temu przez firmę Datel i, mimo swej powolności spełnia całkiem dobrze swoje zadanie — produkuje niemal stuprocentowo dokładne kopie.

NASZ WERDYKT

Polecam Amiga Action Replay III wszystkim użytkownikom Amigi, którzy nie chcą zakończyć swojej przygody z tym komputerem wyłącznie na jego posiadaniu. AR3 jest prosty w obsłudze i daje użytkownikom wiele cennych możliwości; mogę więc z czystym sumieniem polecić ten moduł wszystkim posiadaczom Amigi.

RAFAŁ WIOSNA

DYSTRYBUTOR:
InterComp, ul. Karowa 18a/20 00-324 Warszawa, tel. (022) 26-66-07.

CENA: 1 600 000 zł (maj 1992).

ZALETY
+ wszechstronność zastosowań
+ łatwość obsługi i instalacji
+ dobrze opracowana instrukcja
+ gwarancja na terenie Polski
+ bardzo duże możliwości (w tym wiele unikalnych)

WADY
– niemożność przyłączenia do Amigi modułu i dysku twardego w tym samym czasie
– niektóre polecenia są bardzo długie i niewygodne we wpisywaniu
– stosunkowo wysoka cena

Z OSTATNIEJ CHWILI:
Tuż przed oddaniem numeru do druku dotarła do mnie informacja, że AR3 jest już sprzedawany z polską instrukcją obsługi.

Z NOTATNIKA RED. CHOCHLIKA

■ W "C&A" 03/92 na stronie 19 do programu JOY-KURSOR wkradł się błąd - są dwie linie 255 (druga po wierszu 265). Wprowadź raz jeszcze linię 255:

```
255 d=d+1:c=c+a:goto 210
```

a jej "sobowtóra" po wierszu 265 skasuj.

■ Kolejny błąd dopadł nas w numerze 04/92 w programie EDIT 64, wiersz 160. Zamiast:

```
160 if scs then print "niedobre dane w bloku"w:end
```

powinno być:

```
160 if sc<>s then print "niedobre dane w bloku"w:end
```

Serdecznie przepraszamy Czytelników za błędy i jednocześnie dziękujemy panu Tomaszowi Rutkowskiemu za szybką reakcję. Co do Pana pytania: nie podał Pan niestety swojego adresu !

(KAD)

STACJE TOMS A-880
SPECIAL i STANDARD

Już po kilku tygodniach użytkowania Amigi, każdy przekona się, że 512 KB pamięci i jedna stacja dysków to stanowczo za mało. Szybko więc dokupujemy rozszerzenie pamięci, po czym zwykle pojawia się dylemat: kupić twardy dysk czy zewnętrzną stację dysków. To drugie rozwiązanie, ze względu na swoją cenę jest jak dotychczas znacznie bardziej popularne.

BLASKI STACJI ZEWNĘTRZNYCH 5.25"

Po pierwsze: stosowane w nich dyskietki są i najprawdopodobniej pozostaną tańsze od dyskietek 3.5". Jeśli więc ktoś chce kolekcjonować programy, dużo taniej będzie stosować „duże" dyskietki.

Po drugie: właścicielom stacji 5.25" znacznie łatwiej będzie wymieniać dane z komputerami standardu PC. Ta zaleta powoli co prawda niknie, bowiem we wszystkich niemal sprzedawanych obecnie „pecetach" montowane są już 3.5-calowe stacje dysków, nadal jednak sporo jest użytkowników starych XT czy AT wyposażonych tylko w stację 5.25 cala. Dzięki specyficznej konstrukcji sterownika Amigi, a także programom, istnieje możliwość zapisu i odczytu dyskietek w formacie IBM PC (360 lub 720 KB) za pomocą zwykłej „amigowskiej" stacji.

Po trzecie: zewnętrzna stacja, oprócz zwiększonego komfortu pracy (lub zabawy) z wielodyskowymi programami, umożliwia też nieporównanie łatwiejszą pracę z systemem operacyjnym Amigi. Jest on bowiem tak skonstruowany, że do wykonania najprostszej czynności (nawet wyświetlenia katalogu dyskietki) potrzebny mu jest odpowiedni program z dyskietki systemowej. Oprócz tego, z dyskietki tej wczytywane jest całe multum programów obsługi (np. handlery), a także tzw. biblioteki. Wszystko to sprawia, że co chwila musimy zmieniać dyskietkę w stacji, by dostarczyć systemowi niezbędnych plików. Problem znika z chwilą zastosowania dodatkowej stacji dysków, w której umieszczamy dyskietkę systemową; do pracy wykorzystujemy drugą stację.



KONSTRUKCJA

Obie stacje zewnętrznie różnią się od siebie niewiele. Napędy umieszczono w dwuczęściowych stalowych obudowach pomalowanych na typowy dla Amigi kremowy kolor. Jako kable połączeniowe zastosowano taśmy, co uważam za posunięcie nader udane — połączenie stacji z komputerem jest bardzo elastyczne i nie ma kłopotów ze sztywnym, przeszkadzającym na biurku kablem. Od strony wykonania nie można stacjom nic zarzucić. W obu stacjach zastosowano takie same napędy, obudowy i kable połączeniowe.

Stacje TOMS są sprzedawane jak na razie w dwóch wersjach: SPECIAL i STANDARD. Stacja SPECIAL jest wyposażona w następujące wyposażenie dodatkowe w stosunku do stacji STANDARD:

1) Wyłącznik stacji
2) Złącze do przyłączenia następnej stacji dysków
3) Zabezpieczenie antywirusowe

GWARANCJA I SERWIS

Producent udziela dwunastomiesięcznej gwarancji na swoje wyroby pod warunkiem, że plomby gwarancyjne nie zostaną naruszone. Osobom spoza Warszawy (po uprzednim umówieniu się) producent naprawia stacje na poczekaniu pod warunkiem oczywiście, że taka naprawa jest możliwa do przeprowadzenia.

W AKCJI

Stacja STANDARD jest modelem bardzo prostym i daje się określić pojęciem MINIMUM. Producent nie wyposażył jej w wyłącznik, blokadę antywirusową ani w złącze do przyłączania kolejnych stacji — za to cena tego urządzenia jest bardzo niska i niewiele odbiega od ceny samego napędu. Z kolei model A-880 SPECIAL jest znacznie bogatszy w usprawnienia lecz jest jednocześnie droższy.

WYŁĄCZNIK STACJI

Wyłącznik ten odcina po prostu jedną z linii dzięki czemu Amiga przestaje „widzieć" stację bez konieczności fizycznego odłączenia stacji od komputera. Trzeba bowiem wiedzieć, że każda aktywna stacja to strata w przybliżeniu 20 KB CHIP-RAM. Po podłączeniu dodatkowej stacji dysków, może się więc zdarzyć, że programom wykorzystującym do maksimum minimalną konfigurację Amigi zabraknie pamięci.

ZŁĄCZE DO PRZYŁĄCZENIA KOLEJNEJ STACJI ZEWNĘTRZNEJ

Ta „przelotka" pozwala na dołączenie maksymalnie trzech zewnętrznych stacji dysków. Jej brak w modelu STANDARD muszę niestety uznać za pewną wadę, gdyż w praktyce uniemożliwia to podłączenie do Amigi następnej 3.5-calowej stacji dysków. Aby móc wczytywać programy bezpośrednio z dyskietek o średnicy 5.25", musimy skorzystać z tzw. boot-selectora, który działa tylko dla pierwszej zewnętrznej stacji dysków — nie można więc podłączyć stacji TOMS STANDARD jako drugiej lub dalszej w szeregu. Wada ta oczywiście może nie mieć znaczenia dla osób, które nie myślą o dalszej rozbudowie zestawu.

BLOKADA ANTYWIRUSOWA

Jest dość specyficzne i bardzo ciekawe zabezpieczenie: uniemożliwia ono zapis CZEGOKOLWIEK na dyskietce znajdującej się w stacji. Blokada ta aktywizowana jest zawsze po włączeniu lub inicjalizowaniu (reset) komputera. Takie zabezpieczenie jest w stanie wyeliminować kilka starszych typów wirusów, jednakże nowsze okazy mogą zainfekować dyskietkę

w chwili zapisu danych, stąd nie jest to zabezpieczenie stuprocentowo pewne.

Z przodu stacji jest umieszczona specjalna dioda, informująca o stanie zabezpieczenia: jeśli świeci się światłem ciągłym — zabezpieczenie jest włączone, jeśli przerywanym — zabezpieczenie jest wyłączone i o możliwości zapisu decyduje tylko stan dyskietki. Przy okazji dioda jest też praktyczną sygnalizacją włączenia stacji.

PRZEŁĄCZNIK ŚCIEŻEK

Żadna ze stacji nie jest wyposażona w przełącznik 80/40 ścieżek, ani w mechanizm niwelujący „klikanie" pustej stacji. Brak przełącznika ścieżek nie jest już właściwie wadą, bowiem nowsze wersje programów do przenoszenia danych potrafią odczytywać i zapisywać dyskietki 360 KB także za pomocą stacji 80. ścieżkowej. Poza tym, przy pomocy programu „800" można zmusić" „pecety" do zapisu i odczytu dyskietki 5.25 cala sformatowanej na 720KB.

WSPÓŁPRACA Z OPROGRAMOWANIEM

Amiga jest tak skonstruowana, że przeznaczona dla niej zewnętrzna stacja dysków nie musi zawierać tak wielu układów jak, dajmy na to, stacje dla Commodore 64 czy „małego" Atari. Sterownik stacji znajduje się w komputerze, a niezbędny do podłączenia napędu interfejs jest śmiesznie prosty. Nie ma więc takich problemów, jak na przykład niezgodność ROM (którego w stacjach Amigi oczywiście nie ma). Tak więcc dość trudno w przypadku Amigi mówić o kompatybilności stacji. Jedyne, co może spowodować, odmowę współpracy, to zastosowanie kiepskiego napędu, zbyt wolno reagującego na polecenia, w wyniku czego gorzej napisane programy mogą sobie z nim nie radzić. Taka możliwość istnieje jednak tylko w przypadku programów obsługujących stację z pominięciem systemu operacyjnego — nie ma więc groźby niepoprawnej pracy programów użytkowych. Pozostaje mi więc tylko stwierdzić, że obie stacje pracowały poprawnie i nie znalazłem programu, który nie chciałby działać tylko z nimi.

NASZ WERDYKT

Stacje firmy TOMS są urządzeniami bardzo dobrze prezentującymi się na tle produktów innych polskich producentów tego typu sprzętu. Ich cena jest stosunkowo niska (choć nadal uważam, że za tak proste urządzenie, jakim jest w naszych czasach stacja dysków — za wysoka), a przy tym są bardzo dobrze wykonane.

ANDRZEJ BOBEK

STACJA TOMS A 880 SPECIAL

NAPĘD: Teac
CENA: 1300 tys. zł (kwiecień 1992)
WYMIARY: 245mm x 153mm x 55 mm

ZALETY:
* niska cena
* przydatne usprawnienia
* solidne wykonanie

WADY:
* brak układu tłumiącego „klikanie" stacji

STACJA TOMS A 880 STANDARD

NAPĘD: Teac
CENA: 800 tys. zł (kwiecień 1992)
WYMIARY: 245 x 153 x 55 mm

ZALETY:
* niska cena
* solidne wykonanie

WADY:
* brak układu tłumiącego „klikanie" stacji
* brak złącza umożliwiającego dołączenie kolejnych napędów

PRODUCENT:
TOMS, ul. Widok 14/1, 00-023 Warszawa, tel. 27-18-01 lub 641-54-29 w godzinach 9.00–17.00





POLSKIE ZNAKI dla drukarki STAR LC-20

OD REDAKCJI:
Opisane w tym artykule programy są dostępne na dyskietce 3.5" dla Amigi. Szczegółowe informacje dotyczące zamawiania dyskietek jak również numerów archiwalnych są podane w tym numerze „C&A"

Z problemem jak zmusić drukarkę aby pisała po polsku styka się chyba każdy jej posiadacz. Tak było i w moim przypadku. Początkowo używałem PageSettera z polskim zestawem znaków (wydruk w trybie graficznym). Niestety podczas przygotowywania pewnego artykułu, upstrzonego wzorami pojawiły się trudności, m.in. brak takich funkcji jak (tryb indeksowy i potęgowy subscript i superscript) itp. Zastąpienie ich wymagało definiowania dużej liczby tzw. boxów co komplikowało późniejsze korekty tekstu, jak dopisanie linii czy przeniesienie tekstu na następną stronę. Postanowiłem więc zaprojektować własne znaki by móc je wykorzystać do drukowania w innych edytorach.

Chciałbym tu przedstawić sposób w jaki projektuje się znaki oraz trzy programy, a właściwie zbiory danych, definiujące polskie znaki w trybie draft (o wielkości 10 i 12 CPI) i NLQ. Artykuł ten jest przeznaczony dla zainteresowanych projektowaniem i uzyskiwaniem na wydruku własnych znaków (np. litery greckiego alfabetu, znaki sumy czy iloczynu itd.) i mających trudności z obcojęzycznymi instrukcjami. Programy są przeznaczone dla drukarki LC-20; wykorzystanie ich do innych drukarek jest możliwe po sprawdzeniu zgodności kodów sterujących.

PROJEKTOWANIE ZNAKÓW DRAFT

Choć drukarka LC-20 jest dziewięcioigłowa użytkownik może wykorzystać tylko osiem igieł jednocześnie. Tak więc matryca znaku jest prostokątem o wysokości 8 punktów i szerokości jedenastu. Do projektowania używa się matrycy o wymiarach 8x6 kratek, na której litery ą i ś wyglądają tak jak to przedstawiono na rysunku 1.

Zakreślone punkty to miejsca w które uderzają igły głowicy drukarki. Punkty te mogą być umieszczone wewnątrz kratek bądź na liniach oddzielających poszczególne kratki matrycy. Pamiętaj jednak, że punkty w jednej linii poziomej nie mogą ze sobą sąsiadować tzn. jeżeli jeden punkt znajduje się wewnątrz kratki, to następny nie może znaleźć się na linii oddzielającą tę kratkę od sąsiedniej. Najbliższe wolne miejsce to wnętrze kratki położonej obok.

Teraz trochę arytmetyki. Musimy obliczyć wartości bajtów danych przesyłanych do drukarki dla znaku. Operacja ta przypomina zamianę liczby binarnej na dziesiętną. Każdy punkt reprezentuje bit o wartości 1. Dla ułatwienia obliczeń z lewej strony rysunku pokazane są dziesiętne wartości bitów. Jeżeli w jednej kolumnie matrycy są dwa lub więcej punktów to wartości bitów są sumowane. Dla pierwszej przykładowej litery wygląda to tak:

0,8,20,64,20,64,20,66,57,4,0

Pozostał jeszcze do obliczenia bajt kontrolny, będący pierwszym bajtem danych, i który oblicza się według wzoru:

Last + Left space * 16 + Descender * 128

Last i Left Space określają położenie znaku wewnątrz matrycy, i tak:

Last
Numer ostatniej kolumny, którą zajmuje znak w zakresie 4—11.

Left space
Numer pierwszej kolumny, którą zajmuje znak w zakresie 0—7.

Descender
Określa czy do drukowania będzie wykorzystana dolna, czy górna igła. Dzięki temu znak może być przesunięta bardziej w dół lub w górę, w zależności od potrzeb. Na przykład: gdy chcesz ogonek litery „ą" drukować dwoma punktami, a jednocześnie litera ma być na tym samym poziomie (na tej samej linii bazowej) co pozostałe znaki, bit Descender ustaw (dla „ą" i „ę") na 0, dla innych znaków zaś — na 1. W przeciwnym wypadku litery te będą nienaturalnie przeniesione do góry.

Teraz, gdy już wiemy co nieco o projektowaniu, nadszedł czas na zaprezentowanie dwóch pierwszych programów. Jak widać programy te są zapisane w postaci szesnastkowej; do ich wpisania wykorzystałem program o nazwie filemaster. Najpierw utworzyłem dwubajtowy zbiór za pomocą edytora tekstu, a później zwiększałem jego rozmiar FileMasterem wpisując kolejne liczby. Całość zapisałem na dyskietkę jako plik. Można też skorzystać z programu MasterSeka (dla początkujących: po uruchomieniu wprowadź:

c ‹ENTER›
100 ‹ENTER›

następnie poleceniem „m adres" (np. $40000) wprowadź dane i poleceniem „wi nazwazbioru" zapisz pamięć od adresu $4000 do długość w bajtach + 1) Długości dziesiętne zbiorów są następujące:

DRAFT 10 CPI: 322
DRAFT 12 CPI: 322
NLQ : 950

UWAGA:
Mikrowyłącznik 2-1 drukarki powinien być ustawiony w pozycję OFF (DOWNLOAD), co można wykonać TYLKO WTEDY, GDY DRUKARKA JEST WYŁĄCZONA Z SIECI !!!.

Jak widać z programu 1 kody są zgodne ze standardem ks. Pikula (patrz C&A 01/92). Myślę, że kody polskich znaków w tym standardzie są wybrane dobrze, (w przeciwieństwie do spolszczonego PageSettera, w którym polskie znaki przypisano nawiasom kwadratowym i sześciennym). Program 2 zawiera zestaw znaków draft o wielkości 12 CPI.

Ponieważ po inicjalizacji drukarka jest ustawiona na zestaw znaków draft o wielkości 10 CPI, dlatego przed kopiowaniem należy ustawić rozmiar czcionki na 12 CPI, aby uniknąć sytuacją, że nasze znaki są rozmiaru 12 CPI (tak je zaprojektowaliśmy), a skopiowane — 10 CPI.

PROJEKTOWANIE ZNAKÓW NLQ (krój Courier).

Z tymi znakami sprawa jest trudniejsza — są one drukowane w dwóch przejściach głowicy, przy czym przed drugim przejściem wałek przesuwa się, i następny przebieg jest drukowany o pół punktu niżej od poprzedniego. Matryca znaku dla obu przebiegów ma wielkość 8x32 punktów, czyli siatkę 8x12. Przykładowe litery przedstawiono na rysunku 2; po drugim przejściu otrzymujemy znak pokazany na rysunku 3.
Sam program niestety, ze względu na objętość, jest dostępny jedynie na dyskietce.

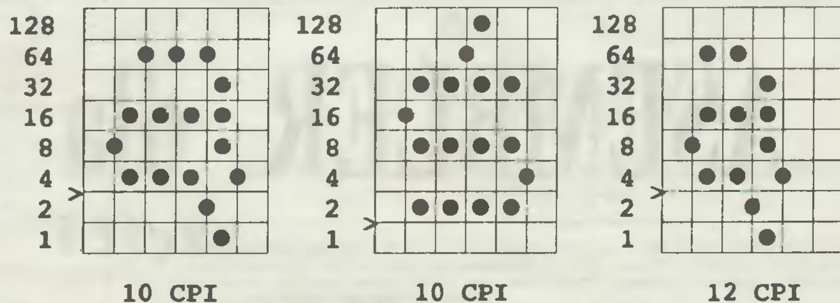
JAK KORZYSTAĆ Z PROGRAMÓW

Jest to stosunkowo proste — wystarczy wysłać po prostu dane do drukarki za pomocą polecenia COPY z poziomu CLI, np.:
COPY PL.NLQ PAR:

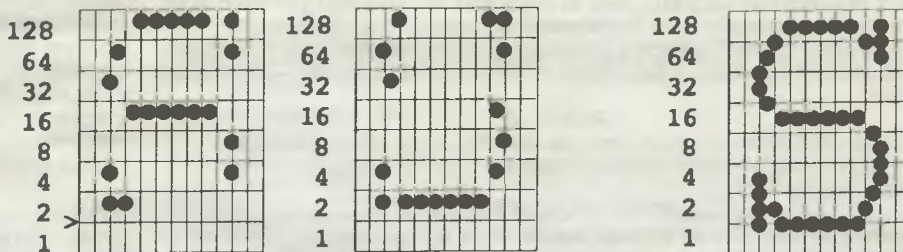
Możliwe jest również umieszczenie tego polecenia w zbiorze startup-suquence pamiętając o włączeniu drukarki PRZED uruchomieniem komputera i przygotowaniu jej do pracy (wyłącznik ON-LINE w pozycję ON). Teraz, gdy chcesz wydrukować tekst z polskimi znakami wyślij go również do PAR:. Aby uniknąć kłopotów z brakującymi plikami obsługi portu równoległego początkujący powinni uruchomić system z dyskietki zawierającej Workbench.

PAWEŁ GALAS

Rysunek:
1. ***Tworzenie znaków w trybie DRAFT***
2. ***Tworzenie znaków w trybie NLQ***
3. ***Znaki NLQ po dwóch przejściach głowicy***



rys. 1



przebieg 1

przebieg 2

rys. 2

litera po dwóch przebiegach

rys. 3

```
PL—DRUK.10cpi

1B 40                ;inicjalizacja drukarki
1B 78 30             ;włączenie trybu pracy draft
1B 3A 00 00 00       ;kopiowanie znaków z ROM do RAM
1B 25 31             ;korzystanie ze znaków download
                      dozwolone
1B 26 00             ;sekwencja definująca nasze znaki
C6 C6                ;kod definiowanego znaku (wysłanie
                      kodu "C6" — spowoduje wydruk "Ą")
8B                   ;bajt kontrolny (Left space = 0,
                      Last = 11, Descender = 1)
00 1E 20 48 80 08 80 48 21 1E 00  ;dane dla "Ą"
1B 26 00 E6 E6 0B ;bajt kontrolny (różni się o bajt
                  ;Descender = 0
00 08 14 40 14 40 14 42 39 04 00 ;dane dla "ą"
1B 26 00 F5 F5 8B
00 10 2A 00 2A 40 AA 00 2A 04 00 ;dane dla "ś"
1B 26 00 D5 D5 8B
00 24 52 00 52 80 52 00 52 0C 00 ;dane dla "Ś"
1B 26 00 E7 E7 8B
00 1C 22 00 22 40 A2 00 22 00 00 ;dane dla "ć"
1B 26 00 FE FE 8B ;bajt kontrolny
00 00 22 04 22 88 22 10 22 00 00 ;dane dla "ż"
1B 26 00 C7 C7 8B
00 3C 42 00 62 00 C2 00 42 24 00 ;dane dla "Ć"
1B 26 00 DE DE 8B
00 00 42 04 4A 80 52 20 42 00 00 ;dane dla "Ż"
1B 26 00 EA EA 0B
00 38 44 10 44 10 44 12 45 30 00 ;dane dla "ę"
1B 26 00 FB FB 8B
00 00 22 04 22 48 A2 10 22 00 00 ;dane dla "ź"
1B 26 00 CA CA 8B
00 FE 00 92 00 92 00 92 01 82 00 ;dane dla "Ę"
1B 26 00 DB DB 8B
00 00 82 14 8A 10 A2 50 82 00 00 ;dane dla "Ź"
1B 26 00 EE EE 8B
00 00 00 8A 00 FE 00 22 00 00 00 ;dane dla "ł"
1B 26 00 F1 F1 8B
00 00 3E 00 20 40 A0 00 20 1E 00 ;dane dla "ń"
1B 26 00 F3 F3 8B
00 1C 22 00 22 40 A2 00 22 1C 00 ;dane dla "ó"
1B 26 00 CE CE 8B
00 FE 00 12 20 42 00 02 00 02 00 ;dane dla "Ł"
1B 26 00 D1 D1 8B
00 FE 00 40 20 50 88 04 00 FE 00 ;dane dla "Ń"
1B 26 00 D3 D3 8B
00 3C 42 00 62 00 C2 00 42 3C 00 ;dane dla "Ó"
1B 50                            ; rozmiar 10cpi
07                               ; sygnał dźwiękowy
```

```
PL—DRUK.12cpi

1B 40                ;inicjalizacja drukarki
1B 78 00             ;właczenie trybu pracy draft
1B 4D                ;ustawienie 12cpi
1B 3A 00 00 00       ;kopiowanie znaków z ROM do RAM
1B 25 01             ;korzystanie ze znaków download
                     ;dozwolone
1B 26 00 C6 C6 8B
00 1E 20 48 80 48 21 1E 00 00 00 ; dane dla "Ą"
1B 26 00 E6 E6 0B
00 08 54 00 54 02 39 04 00 00 00 ; dane dla "ą"
1B 26 00 F5 F5 8B
00 10 2A 00 6A 80 2A 04 00 00 00 ; dane dla 'ś"
1B 26 00 D5 D5 8B
00 24 52 00 D2 00 52 0C 00 00 00 ; dane dla "Ś"
1B 26 00 E7 E7 8B
00 1C 22 00 62 80 22 00 00 00 00 ; dane dla "ć"
1B 26 00 FE FE 8B
00 22 04 22 88 22 10 22 00 00 00 ; dane dla "ż"
1B 26 00 C7 C7 8B
00 3C 42 20 42 80 42 24 00 00 00 ; dane dla "Ć"
1B 26 00 DE DE 8B
00 42 04 4A 80 52 20 42 00 00 00 ; dane dla "Ż"
1B 26 00 EA EA 0B
00 38 44 10 44 12 45 30 00 00 00 ; dane dla "ę"
1B 26 00 FB FB 8B
00 22 04 22 48 A2 10 22 00 00 00 ; dane dla "ź"
1B 26 00 CA CA 8B
00 FE 00 92 00 92 01 82 00 00 00 ; dane dla "Ę"
1B 26 00 DB DB 8B
00 82 14 8A 10 A2 50 82 00 00 00 ; dane dla "Ź"
1B 26 00 EE EE 8B
00 00 8A 00 FE 00 22 00 00 00 00 ; dane dla "ł"
1B 26 00 F1 F1 8B
00 00 3E 00 60 80 20 1E 00 00 00 ; dane dla "ń"
1B 26 00 F3 F3 8B
00 1C 22 00 62 80 22 1C 00 00 00 ; dane dla "ó"
1B 26 00 CE CE 8B
00 FE 00 12 20 42 00 02 00 00 00 ; dane dla "Ł"
1B 26 00 D1 D1 8B
00 BE 40 20 50 88 04 FA 00 00 00 ; dane dla "Ń"
1B 26 00 D3 D3 8B
00 3C 42 20 42 80 42 3C 00 00 00 ; dane dla "Ó"
07                               ; sygnał
```

ASEMBLER dla KAŻDEGO

część I

W ciągu miesiąca miałeś chyba wystarczająco dużo czasu, by przynajmniej zacząć kompletować oprogramowanie i książki o którym pisałem w poprzednim odcinku. Jeśli dysponujesz już omawianym wyposażeniem możemy zabrać się za ćwiczenia praktyczne.

Pierwszym celem, jaki sobie postawimy, będzie wyświetlenie na ekranie literki „A". Za pomocą BASIC zrobić to można bardzo prosto — wystarczy napisać:

PRINT „A"

i wcisnąć RETURN. „A" na pewno się pokaże.
No dobrze, ale jeśli chcemy mieć „A"w lewym górnym rogu?
Proszę bardzo:

PRINT CHR$ (147); „A"

Bardzo ładnie. Skrzaty mówią jednak, że w komputerze Commodore 64 można zrobić wszystko za pomocą jednej lub kilku instrukcji POKE. Wpisz więc:

POKE 1024,1

Tak naprawdę BASIC stanowi jedynie ogniwo pośrednie pomiędzy użytkownikiem i asemblerem. Komputer nie jest w stanie zrozumieć kierowanych do niego angielskich słów PRINT, POKE czy LIST bez odpowiedniego programu w asemblerze. Wynika z tego, że program na wyświetlenie literki „A" napisany w BASIC został, PRZED wykonaniem, przetłumaczony na język maszynowy. Czy więc nie byłoby prościej dać komputerowi rozkazy od razu w asemblerze? Otóż byłoby!

Uruchom teraz jeden ze swoich programów (monitor, asembler). Jeśli masz monitor, to wpisz:

A 2710 LDA #$01
A 2712 STA $0400
A 2715 RTS

Teraz wpisz G 2710 i wciśnij RETURN. Jeśli dysponujesz programem TurboAssembler wpisz:

*** = $2710**
LDA #$01
STA $0400
RTS

Naciśnij teraz strzałkę w lewo, trójkę, a potem „S".

Obie operacje powinny dać taki sam efekt: w lewym górnym rogu ekranu pojawi się literka „A". Jeśli niezbyt podoba się nam „A", możemy ją zastąpić dowolnym innym znakiem, wpisując zamiast LDA #$01, np. LDA #$02 (litera B), LDA #$03 (C) i tak dalej. Jeśli nie podoba Ci się miejsce w którym jest wyświetlany Twój znak spróbuj zmodyfikować wiersz zawierający instrukcję STA $0400.

Tu należy się Tobie słowo wyjaśnienia na temat struktury ekranu na którym pojawia się tekst. Zastanów się nad następującym zagadnieniem: włączysz komputer, wpisujesz jakiś tekst, powiedzmy „ZABAWA Z KOMPUTEREM". O ile w domu nie wyłączą prądu tekst ten będzie na ekranie widoczny tak długo jak komputer będzie włączony, chociaż ekran telewizora czy monitora na pewno NIE JEST wyposażony w pamięć. Skąd więc komputer wie GDZIE i JAKI tekst na ekranie należy umieścić?

Rozwiązano to niezwykle prosto: dane przeznaczone do wyświetlenia na ekranie są najpierw zapisywane w specjalnym obszarze pamięci określanej zwykle mianem „pamięci ekranowej". Stamtąd są one pobierane i po przetworzeniu wysyłane na ekran telewizora. Wniosek z tego jest prosty: tekst można wyświetlać tak długo jak długo jest on zawarty w pamięci ekranowej komputera. Podobnie jest z grafiką i kolorami, chociaż dla nich zarezerwowano inne obszary pamięci, mają one również różną wielkość (ok. 1 KB dla ekranu tekstowego lub 8 KB dla grafiki).

Obszar pamięci ekranowej (tzn. pamięci przeznaczonej do przechowywania TEKSTU) jest zlokalizowany w Commodore 64 od adresu 1024 do 2023 co w systemie szesnastkowym odpowiada adresom $0400 do $07e7. Zwróć uwagę, że adres $0400 wymieniony w drugim wierszu programu odpowiada dokładnie komórce o adresie 1024 będącej pierwszą komórką pamięci ekranu tekstowego. Zmiana adresu $0400 powinna więc spowodować wyświetlenie litery „A" w innym miejscu ekranu. Spróbujmy teraz przyjrzeć się uważnie naszemu programikowi.

LDA jest mnemonikiem czyli trzyliterowym skrótem (synonimem) rozkazu LoaD Accumulator (załaduj akumulator). Wymieniona za nim liczba (zwana argumentem) jest, w tym wypadku, wartością do wczytania do akumulatora.

STA jest mnemonikiem rozkazu STore Accumulator (zapisz zawartość akumulatora). Rozkaz ten powoduje zapisanie liczby wpisanej uprzednio do akumulatora w miejscu określonym liczbą wpisaną bezpośrednio po STA. W naszym programie będzie to komórka pamięci o adresie $0400 (1024 dziesiętnie).

Rozkaz **RTS** umożliwia powrót do BASIC zaraz po zakończeniu naszego programu. Jest to jeden z rozkazów nie wymagających podawania argumentów.

Jeśli zajrzysz do spisu rozkazów asemblera, zobaczysz:

LDA — LoaD Accumulator (załaduj akumulator)
STA — STore Accumulator (zapisz akumulator)
RTS — ReTurn from Subroutine (powróć z podprogramu)

Możemy z tego wywnioskować, że miejsce w pamięci, do którego komputer wpisał liczbę $01 (1 dziesiętnie), nazywa się akumulatorem. Akumulator może przechowywać tylko jedną liczbę; jeśli wpiszesz doń nową liczbę, stara zawartość akumulatora zostanie bezpowrotnie skasowana.

No dobrze. O rozkazach, których używamy, wiemy już nieco więcej. Wiemy też, czego się po nich spodziewać. Wpiszmy więc:

Monitor	Asembler
	* = $2710
A 2710 LDA #$01	LDA #$01
A 2712 STA $0400	STA $0400
A 2715 LDA #$02	LDA #$02
A 2717 STA $0401	STA $0401
A 271A LDA #$03	LDA #$03
A 271C STA $0402	STA $0402
A 2720 RTS	RTS
G 2710	Strzałka w lewo „3", „a"

Teraz na ekranie powinien się pokazać ciąg liter: A, B. C. I to wszystko za pomocą dwóch, najprostszych rozkazów asemblera. Aby nabrać wprawy, proponuję jeszcze trochę samemu poeksperymentować. Spróbuj na przykład wyświetlić zamiast liter „ABC" litery X, Y i Z w drugim wierszu na ekranie (w jednym wierszu mieści się 40 znaków).

Po jakichś trzech–czterech próbach wyświetlania liter, proponuję coś bardziej rozbudowanego: oprócz litery, zmienić jeszcze jej kolor. Wiemy już, że kolory są przechowywane, podobnie jak tekst, w określonym obszarze pamięci RAM komputera; pamięć tę nazywamy pamięcią koloru. Mieści się ona w obszarze od $d800 a $dbff (55296 – 56319 dziesiętnie).

Jak jednak obliczyć numer komórki, do której wstawimy kolor? To stosunkowo proste — od adresu komórki pamięci ekranowej, w której zawarta jest nasza litera (np. $0428) odejmij $0400 (zostanie $28), a potem dodaj $d800 w wyniku czego otrzymasz $d828. Wartość zapisana w tej komórce ($d828) decyduje o kolorze znaku zapisanego w pamięci ekranowej (komórka $0428). Jeśli masz problemy z przeliczaniem adresów w systemie szesnastkowym to podaję adresy w postaci dziesiętnej:

Pamięć ekranowa	Pamięć koloru
$0400 = 1024	$d800 = 55296
+$0028 = 40	+ $0028 = 40
$0428 = 1064	$d828 = 55336

Jeżeli kod szesnastkowy nie chce Ci wejść żadną miarą do głowy, to może warto byłoby zaopatrzyć się w jakiś podręczny kalkulator mający możliwości przeliczania wartości w trzech systemach: dziesiętnym, szesnastkowym i binarnym. Teraz wpisz i spróbuj uruchomić ten oto program:

Monitor	Asembler
	* = $2710
A 2710 LDA #$01	LDA #$01
A 2712 STA $0428	STA $0428
A 2715 LDA #$00	LDA #$00
A 2717 STA $d828	STA $d828
A 271A RTS	RTS
G 2710	Strzałka w lewo
	„3", „a"

Mamy już czarną literę „A"! Może być też biała, czerwona — jaką tylko chcemy. Trzeba w tym celu zmienić wartość argumentu po rozkazie LDA w linii 2715.

BARTŁOMIEJ I. KACHNIARZ

Uwaga
Do wpisywania wszystkich zamieszczonych w tym artykule NIEZBĘDNY jest program typu MONITOR lub ASEMBLER."

Dziś poznaliśmy instrukcje:
LDA #xx — wstaw wartość xx do akumulatora
STA xxxx — zapisz akumulator do komórki o adresie xxxx
RTS — powróć z podprogramu

UKŁAD VIC

VIDEO INTERFACE CHIP — cz. II

53273 ($d019)
Zezwolenie na przerwania IRQ wywoływane przez układ VIC.

Bit 0:
1 = dojście do określonej w 53266 ($d012) linii rastra powoduje przerwanie IRQ
0 = raster nie ma wpływu na przerwania

Bit 1:
1 = zderzenie duszka z tłem powoduje przerwanie IRQ
0 = zderzenia duszków z tłem nie mają wpływu na przerwania

Bit 2:
1 = zderzenie dwóch duszków powoduje przerwanie IRQ
0 = zderzenie dwóch duszków ze sobą nie ma wpływu na przerwania

Bit 3:
1 = wciśnięcie przełącznika pióra świetlnego powoduje wytworzenie przerwania IRQ
0 = pióro świetlne nie wpływa na generację IRQ

Bity 4–6: Niewykorzystane.

Bit 7:
1 = układ VIC ma prawo wywoływać przerwania IRQ
0 = układ VIC nie wywołuje przerwań IRQ

Normalnie, przerwanie IRQ wywołuje zegar komputera, co 1/50 sekundy. Można jednak zastrzec, by przerwanie to wywoływał układ VIC. VIC może wywołać IRQ gdy:
— rejestr rastra osiągnie zadaną wartość (bit 0),
— duszek zakryje jakąś część ekranu graficznego (bit 1),
— duszek zakryje fragment innego duszka (bit 2),
— operator wciśnie przełącznik pióra świetlnego lub przycisk FIRE joysticka w porcie 1 (bit 3).

Uwaga:
Jeśli powodem przerwania są zderzenia duszków, nie są istotne ich prostokątne matryce. Dla VIC ważna jest jedynie ich faktyczna zawartość. Nawet tylko jeden punkt duszka jest w stanie wywołać „zderzenie" i w efekcie przerwanie IRQ.
Aby zapobiec powtórnemu wywołaniu przerwania z tego samego powodu komputer zeruje odpowiedni bit. Jeżeli korzystasz np. tylko z przerwań rastra, musisz po każdym wywołaniu przerwania rastra ustawić na 1 bit o numerze 0 w tej komórce. Jeśli tego nie zrobisz, przerwania IRQ wywoływać będzie tylko zegar.
Można zmusić VIC, aby generował przerwanie IRQ z dwóch różnych powodów. Jeśli np. układasz grę i chcesz wywołać przerwanie spowodowane kolizją dwóch duszków, ustaw bit 2. Jeśli jednak chcesz, żeby zostało wywołane przerwanie rastra, ustaw także bit 0. Powodem przerwania staje się to wydarzenie, które nastąpi wcześniej. Jeśli ustawisz którykolwiek z bitów tego rejestru, komputer sam ustawi (1) bit siódmy, na znak, że przerwania będą od teraz wywoływane przez układ VIC.

53274 ($d01a)
Rejestr maski przerwań

Bit 0:
1 = przerwania rastra będą wywoływane
0 = przerwania rastra nie będą wywoływane

Bit 1:
1 = zderzenie duszka z tłem wywołuje przerwanie
0 = zderzenie duszka z tłem nie wywołuje przerwania

Bit 2:
1 = zderzenie dwóch duszków wywołuje przerwanie
0 = zderzenie dwóch duszków nie wywołuje przerwania

Bit 3:
1 = wciśnięcie przycisku pióra wywołuje przerwanie
0 = wciśnięcie przycisku pióra nie wywołuje przerwania

Bity 4–7: Niewykorzystane.

Ten bajt zawiera zezwolenia wywoływania przerwań. Jeżeli chcesz, by komputer przyjął do wiadomości zezwolenia, które wydałeś w rejestrze 53273 ($d019), musisz mu przedtem na to zezwolić w rejestrze 53274 ($d01a). Aby np. komputer uzależnił wywoływanie przerwań od rejestru rastra, musisz najpierw ustawić (1) bit 0 w tej komórce, dopiero potem komórka 53273 ($d019) będzie działać poprawnie. Jeśli wszystkie bity rejestru będą w stanie 0, VIC nie zgłosi żadnego przerwania IRQ. Nie wiadomo, czemu służy takie podwójne zabezpieczenie przerwań przed wywoływaniem ich przez VIC. Na pewno nie wygodzie użytkownika.

53275 ($d01b)
Rejestr pierwszeństwa wyświetlania duszków i grafiki. Poszczególne duszki reprezentowane są kolejnymi bitami tego rejestru. Jeśli dany bit jest ustawiony (1), to odpowiadający mu duszek jest maskowany (zasłaniany) przez obraz graficzny (np. samochód wjeżdża do tunelu). Ustawienie bitu na 0 powoduje efekt odwrotny (duszek maskuje grafikę).
Jeżeli nakażesz komputerowi wyświetlanie w tym samym miejscu grafiki i duszka nastąpiłyby kłopoty. Aby je wyeliminować wymyślono ten właśnie rejestr. Gdy wyświetlasz duszka nr 0 w tym samym

miejscu, co literę „A", komputer nie może jednocześnie pokazać obu rzeczy. Jeśli chcesz widzieć duszka, wyłącz odpowiadający mu bit pierwszeństwa (w tym przypadku — bit o numerze 0). Jeśli natomiast pragniesz zobaczyć widzieć literę należy ten bit ustawić (1). Takich możliwości nie mamy w wypadku duszków. Obowiązuje tu zasada, że ZAWSZE duszek o numerze NIŻSZYM zasłania duszki o numerze wyższym. Oznacza to, że wyświetlając w tym samym miejscu duszki o numerach 3 i 6 duszek 6 będzie zasłonięty duszkiem o numerze 3. Działanie to ilustruje program 7.

Uwaga:
Jeśli pracujesz w trybie wielokolorowym, duszki zawsze będą maskowały kolor określany przez parę bitów 01, niezależnie od pierwszeństwa ich wyświetlania.

53276 ($d01c)
Rejestr ten pozwala włączyć tryb wielokolorowy dla duszków. Poszczególne duszki reprezentowane są przez kolejne bity tego rejestru. Ustawienie bitu (1) powoduje włączenie dla odpowiedniego duszka trybu wielokolorowego. Gdy bit jest wyłączony, duszek jest wyświetlany w trybie normalnym.
W trybie wielokolorowym każdy punkt duszka jest określany przez dwa bity. Powoduje to zmniejszenie rozdzielczości poziomej duszka, tzn. każdy jego punkt będzie teraz dwa razy szerszy, lecz liczba ich zmniejszy się dwukrotnie. Ogólna szerokość duszka nie ulegnie zmianie. Daje to nam zwiększenie (z 2 do 4) liczby kolorów, jakie możemy stosować dla każdego duszka. Kolor par punktów określają komórki:

00 — komórka koloru tła — 53280 ($d020)
01 — komórka 53285 ($d025)
10 — komórka 53287 ($d027)
11 — komórka 53286 ($d086)

53277 ($d01d)
Dwukrotne rozszerzenie duszka w poziomie. Poszczególne duszki reprezentowane są kolejnymi bitami tego rejestru. Działanie to jest zgodne z opisem dla komórki 53271.

Bit 0: poziome rozszerzenie duszka o numerze 0
Bit 1: poziome rozszerzenie duszka o numerze 1
Bit 2: poziome rozszerzenie duszka o numerze 2
Bit 3: poziome rozszerzenie duszka o numerze 3
Bit 4: poziome rozszerzenie duszka o numerze 4
Bit 5: poziome rozszerzenie duszka o numerze 5
Bit 6: poziome rozszerzenie duszka o numerze 6
Bit 7: poziome rozszerzenie duszka o numerze 7

53278 ($d01e)
Komórka informująca o zetknięciu się ze sobą duszków. Poszczególne duszki są reprezentowane przez kolejne bity tego rejestru. Jeśli dany bit = 1, to odpowiedni duszek zderzył się z innym duszkiem. Kolizja jest sygnalizowana, gdy nałożą się na siebie dwa włączone punkty obu duszków. Jeżeli zetkną się punkty wyłączone, kolizja nie jest sygnalizowana.

Uwaga:
Kolor duszka nie gra tu roli.

Jeśli jednocześnie nastąpi zetknięcie się ze sobą więcej niż dwóch duszków nie da się za pomocą tego rejestru ustalić kolejności tych kolizji. Jeśli np. rakieta zderzy się z kosmitą, a inny kosmita — z meteorem, komputer podaje, że zderzyli się dwaj kosmici, rakieta i meteor.

53279 ($d01f)
Komórka sygnalizująca zderzenia duszków z grafiką (tłem). Komputer włącza określony bit tej komórki, jeśli odpowiadający mu duszek (a konkretnie jego włączony punkt) nałożył się na jakiś element grafiki. Przy wyłączonym punkcie zderzenie nie będzie sygnalizowane.

Uwaga:
Kolory duszka ani podkładu graficznego nie mają znaczenia. Jeśli duszek i tło są czarne, ich kolizja będzie sygnalizowana w sposób ciągły.

53280 ($d020)
Rejestr ten określa kolor ramki. Wartość początkowa = 15 (jasnoniebieski). W tym kolorze jest wyświetlana ramka wokół ekranu tekstowego lub graficznego. Jeśli bit 4 komórki 53265 ($d011) będzie równy 0, cały ekran będzie wyświetlany w tym kolorze.

53281 ($d021)
Rejestr określający kolor tła ekranu. Wartość początkowa = 6 (niebieski). Taki kolor będzie miało tło, niezależnie od włączonego trybu graficznego. Jedynym wyjątkiem jest tryb tła wielokolorowego.

53282 ($d022)
Kolor tła nr 1. Wartość początkowa = 1 (biały). W trybie wielokolorowym w tym kolorze są wyświetlane pary bitów 01 (w trybie grafiki wielokolorowej). W trybie wielokolorowego tła kolor ten ma tło znaków o kodach ekranowych od 64 do 127.

53283 ($d023)
Kolor tła nr 2. Wartość początkowa = 2 (czerwony). W trybie wielokolorowym w tym kolorze są wyświetlane pary bitów 10. W trybie wielokolorowego tła kolor ten ma tło znaków o kodach ekranowych od 128 do 191.

53284 ($d024)
Kolor tła nr 3. Wartość początkowa = 3 (turkusowy). W trybie wielokolorowego tła kolor ten ma tło znaków o kodach ekranowych od 192 do 255.

53285 ($d025)
Kolor pary bitów 01 dla duszka wielokolorowego. Wartość początkowa = 4 (purpurowy). W trybie duszków wielokolorowych rejestr ten zawiera wartość dla koloru pary bitów 01.

53286 ($d026)
Kolor pary bitów 11 dla duszka wielokolorowego. Wartość początkowa = 4 (czarny). W trybie duszków wielokolorowych rejestr ten zawiera wartość dla koloru pary bitów 11.

53287 ($d027)
Kolor duszka nr 0. Wartość początkowa = 1 (biały). W trybie normalnym ten rejestr zawiera kolor bitów ustawionych (1). W wypadku duszka wielokolorowego kolor ten przybiorą pary bitów 10. Bity te są pobierane z obszaru pamięci przypisanego danemu duszkowi. Zasada ta obowiązuje dla komórek podanych poniżej:

53288 ($d028) — kolor duszka nr 1 (2 — czerwony)
53289 ($d029) — kolor duszka nr 2 (3 — turkusowy)
53290 ($d02a) — kolor duszka nr 3 (4 — purpurowy)
53291 ($d02b) — kolor duszka nr 4 (5 — zielony)
53292 ($d02c) — kolor duszka nr 5 (6 — niebieski)
53293 ($d02d) — kolor duszka nr 6 (7 — żółty)
53294 ($d02e) — kolor duszka nr 7 (12 — szary)

53295 — 53311 ($d02f — $d3ff)
Ten obszar pamięci nie ma żadnego praktycznego znaczenia. Mimo, że VIC może adresować 64 komórki, wykorzystanych jest jedynie 47. Pozostałe zawierają zawsze wartość 255, niezależnie od tego, co tam zapiszemy.

BARTŁOMIEJ KACHNIARZ

TURBO i COMMODORE 16/116/PLUS/4

Powolna współpraca komputerów firmy Commodore z magnetofonem w roli pamięci zewnętrznej (rozwiązanie popularne ze względu na cenę zestawu) spowodowała powstanie tzw. „systemów turbo", czyli programów przyspieszających operację odczytu i zapisu z taśmy. Dla komputerów C-16, C-116, PLUS/4 takich systemów powstało kilkanaście.

Generalnie istnieją dwa sposoby działania takich programów. Pierwszy polega na całkowitej zmianie procedury zapisu, i odczytu, co powoduje mniej więcej dziewięciokrotne przyspieszenie tych operacji w stosunku do odczytu i zapisu „normalnego". Pociąga to jednak za sobą konieczność obecności programu „turbo" w pamięci przed rozpoczęciem odczytu/zapisu. Sposób ten nie przyjął się dla C-16 z powodu małej pamięci.

Druga metoda polega na zmianie procedury zapisu tak, aby w czołówce nagrywanego programu znalazła się krótka procedura przyspieszająca wczytanie umieszczonego tuż za nią właściwego programu. Dzięki takiemu rozwiązaniu „turbo" musi znajdować się w pamięci komputera tylko w momencie zapisywania programu na taśmie. Program, zapisany w ten sposób wczytujemy za pomocą zwykłego LOAD, a współpraca z magnetofonem i tak przebiega około sześciu razy szybciej.

NOVALOAD

Najpopularniejszy przedstawiciel metody numer dwa. W systemie tym były zapisane niemal wszystkie gry „firmowe" i dzięki temu nie brak najróżniejszych programów kopiujących (a także zabezpieczających przed skopiowaniem) dla tego systemu. NOVALOAD poznajemy po tym, że podczas wczytywania ekran jest wygaszony, na ramce ukazują się różnobarwne paski, a na środku ekranu widnieje tytuł wczytywanej gry.

NOVALOAD charakteryzuje się dość dużą pewnością odczytu. Pozwala on na zapisanie całej pamięci użytkownika łącznie z pamięcią ekranu (tzn. obszar od $0800 do $FCFF). Istnieje możliwość samoczynnego uruchomienia się programu po wczytaniu bez względu na to, czy jest on napisany w asemblerze czy w BASIC.

Po wczytaniu zapisanego w NOVALOAD programu nie jest przywracana normalna transmisja z magnetofonem, tak więc ponowne wydanie polecenia LOAD spowoduje poszukiwanie na taśmie kolejnego bloku programu (aby tego uniknąć, należy przywrócić wektorowi LOAD, czyli komórkom $032E, $032F standardowe wartości). Powyższa cecha jest często wykorzystywana do zapisywania gier złożonych z kilku części.

HEADER-SAVER TURBO

Jakiś czas temu wyszedł z użycia z powodu największej chyba zawodności (wymaga bardzo dokładnego ustawienia skosu głowicy, nie sygnalizuje błędów podczas operacji wczytywania), a na dodatek zakłóca działanie interpretera (nie można zatrzymać programu w BASIC). Plusem jest fakt, że można zapisać na taśmie całą pamięć użytkownika (lecz w przeciwieństwie do NOVALOAD nie można zrezygnować z samoczynnego uruchamiania programu). Nagranie w systemie HS łatwo rozpoznać: podczas wczytywania na całym ekranie są wyświetlane grube pasy (na przemian w kolorze ramki i tła).

TSL (TURBO SAVE-LOAD)

Najwygodniejszy system turbo dla osób piszących własne programy w BASIC. Tworzy on fikcyjny „szybki magnetofon", obsługiwany poleceniem SAVE „nazwa",7 (lub SAVE „nazwa",7,1 jeśli program napisany w BASIC ma się sam uruchomić). Możliwe jest także zapisanie dowolnego obszaru pamięci z poziomu monitora: S„nazwa",07, pocz, kon, gdzie w miejsce pocz i kon wpisujemy szesnastkowe adresy początku i końca zapisywanego obszaru pamięci. System rezyduje w pamięci aż do wyłączenia komputera — dzięki temu możemy korzystać z niego wielokrotnie.

TWIDDY TURBO

Spotykany raczej rzadko i charakteryzujący się tym, że gra (najczęściej kilkuczęściowa) przegrana w innym systemie niż TWIDDY, nie będzie działać. Tak jest na przykład z programem WINTER EVENTS i kilkoma innymi. Pewność wczytywania jest mniej więcej taka sama, jak dla NOVALOAD. Do programu nie dołączono żadnych instrukcji — dlatego informuję, że przyspieszony zapis programu w BASIC następuje po wydaniu polecenia SAVE (oczywiście odpowiednio zmienionej przez procedurę zapisu umieszczoną w pamięci od adresu $3S40 w przypadku C-16 i C-116 lub od adresu $7D40 w przypadku PLUS/4). Zmiana lokalizacji procedury i dołączenie autostartu nie jest trudne, ale wymaga przynajmniej podstawowej znajomości asemblera.

Aby przekonać się, który z w/w systemów jest najefektywniejszy spróbowałem zapisać na kasecie grę EXORCIST o długości 49. bloków na dysku (tj. około 12 kB). Zapis został wykonany w systemach NOVALOAD, HS, TSL, TZ, normalnym (firmowym) i dodatkowo na dysku. Wyniki przedstawię za miesiąc.

Na zakończenie kilka słów o zamieszczonych obok listingach. Pierwszy zawiera program pomagający rozpoznać, w jakim systemie jest zapisana gra. Obsługa sprowadza się do ustawienia taśmy na początek testowanego programu i wciśnięcia klawisza F1. Po odnalezieniu czołówki i wczytaniu jej do pamięci zostaje podany tytuł gry i system, w którym została ona zapisana (lub komunikat o nierozpoznaniu systemu). Przy okazji muszę dodać, że powstało bardzo dużo przeróbek systemu NOVALOAD: różnią się one od siebie zazwyczaj kilkoma bajtami (mającymi na celu utrudnienie kopiowania), lecz mimo to będą one rozpoznane przez nasz program jako NOVALOAD.

Program rozpoznający można umieścić w dowolnym miejscu pamięci podając adres początkowy w linii 10. Drugi zamieszczony tutaj listing był już raz publikowany w Bajtku 11/87 — powtarzamy go, gdyż większość z Was prawdopodobnie nie ma tego numeru. Jest to wydruk TURBO SAVE-LOAD. Przed uruchomieniem program należy koniecznie zapisać, gdyż kasuje się on automatycznie z pamięci. Pytanie o obszar pamięci umożliwi umieszczenie tego kodu w miejscu nie kolidującym z zapisywanymi danymi.

I jeszcze jedno: jeśli chcesz uniknąć straty czasu i błędów we wpisywaniu zamieszczonych obok programów zamów po prostu dyskietkę C&A nr 06/92.

Mr WAX

LISTING 1

```
100 rem ***********************
105 rem *  turbo save-load  *
110 rem ***********************
115 :
120 scnclr:char,2,2,"   gdzie umieszczamy progra
    m ?":print:print:n=2
125 do:printtab(13)n" - "n"d00$":n=n+1:loop unti
    l n*16>peek(56)orn=8
130 getkey m$:m=val(m$):if m<1 or m>n-1 then 130
135 print:print tab(10)"momencik..."
140 pocz=m*4096+3328:ko=m*16+13
145 for i=51 to 55 step 2:poke i,0:poke i+1,ko:n
    ext
150 d=pocz:read q
155 do:c=c+q:if q>60 and q<64 then q=q+(m-3)*16
160 poke d,q:d=d+1:read q:loop untilq<0
165 if d<>pocz+635 then print:print "brakuje dan
    ych !":end
170 if c<>74748 then print:print "niedobre dane
    !":end:elsec=0
175 for i=0 to 3:read d,q:c=c+d+q:poke m*4096+12
    80+d,q:next
180 if c<>549 then print:print "sprawdz linie 99
    9":end
185 scnclr:print" szybkie nagrywanie programu na
    pisanego":print " w basic"
190 print"save"chr$(34)"nazwa"chr$(34)",7"
195 print" save"chr$(34)"nazwa"chr$(34)",7,1 - a
    uto-run"
200 sys pocz:new
205 :
210 data 169,011,141,048,003,169,061,141,049,003
215 data 096,072,165,174,201,007,240,004,104,076
220 data 164,241,165,178,141,123,063,165,179,141
225 data 124,063,165,157,141,125,063,165,158,141
230 data 126,063,165,173,141,191,062,160,015,169
235 data 032,153,192,062,136,016,250,164,171,192
240 data 017,144,002,160,016,136,048,008,177,175
245 data 153,192,062,076,065,061,160,072,185,119
250 data 062,153,255,006,136,208,247,032,025,227
255 data 169,072,141,036,003,169,003,141,037,003
260 data 169,001,170,168,032,186,255,169,188,162
265 data 192,160,062,032,189,255,169,000,133,154
270 data 169,003,133,035,169,036,133,034,169,034
275 data 160,003,162,038,032,216,255,169,075,141
280 data 036,003,169,236,141,037,003,169,000,170
285 data 168,133,178,169,007,133,179,133,158,169
290 data 176,133,157,032,200,061,169,128,133,154
295 data 162,000,160,000,173,123,063,133,178,173
300 data 124,063,133,179,173,125,063,133,157,173
305 data 126,063,133,158,032,200,061,076,003,135
310 data 032,018,062,165,178,032,064,062,165,179
315 data 032,064,062,165,157,032,064,062,165,158
320 data 032,064,062,132,180,164,178,169,000,133
325 data 178,177,178,032,064,062,200,208,002,230
330 data 179,196,157,165,179,229,158,144,238,165
335 data 180,032,064,062,032,084,062,169,027,141
340 data 006,255,141,062,255,169,008,133,001,088
345 data 032,132,255,096,120,141,063,255,169,000
350 data 133,001,169,011,141,006,255,202,208,253
355 data 136,208,250,169,160,141,002,255,169,000
360 data 141,003,255,169,016,141,009,255
365 data 160,064,169,016,032,064,062
370 :
375 data 136,208,248,169,090,133,171,069,180,133
380 data 180,169,008,133,172,038,171,032,084,062
385 data 198,172,208,247,096,162,108,144,002,162
390 data 255,032,093,062,169,016,044,009,255,240
395 data 251,072,169,000,142,002,255,141,003,255
400 data 104,141,009,255,165,001,073,002,133,001
405 data 096,169,072,141,252,255,169,003,141,253
410 data 255,141,063,255,238,025,255,024,032,079
415 data 003,141,062,255,165,032,141,025,255,088
420 data 032,138,255,032,132,255,165,158,197,157
425 data 208,025,173,071,007,240,014,169,255,141
430 data 012,255,141,013,255,032,190,139,076,220
435 data 139,032,157,138,076,003,135,162,029,076
440 data 131,134,000,032,032,032,032,032,032,032
445 data 032,032,032,032,032,032,032,032,032,032
450 data 173,025,255,133,034,056,120,008,032,147
455 data 003,032,184,003,168,169,000,133,178,032
460 data 184,003,133,179,032,184,003,133,045,032
465 data 184,003,133,046,032,184,003,145,178,069
470 data 158,133,158,200,208,005,230,179,238,025
475 data 255,196,045,165,179,229,046,144,231,032
480 data 184,003,133,157,169,008,133,001,040,176
485 data 001,096,076,000,007,169,000,133,001,202
490 data 208,253,136,208,250,132,158,162,000,032
495 data 198,003,038,172,165,172,201,016,208,245
500 data 032,184,003,201,016,240,249,201,090,208
505 data 234,096,169,001,133,172,032,198,003,038
510 data 172,144,249,165,172,096,169,016,036,001
515 data 208,252,036,001,240,252,045,009,255,072
520 data 169,001,162,120,142,002,255,141,003,255
525 data 169,016,141,009,255,173,025,255,024,105
530 data 016,141,025,255,104,010,010,010,010,096
535 data -1
540 data 007,062,020,063,131,063,141,062
```

LISTING 2

```
100 pocz=dec("3f00")
105 for i=pocz to pocz+175
110 read p:poke i,p:z=z+p
115 next i
120 if z<>15945 then print "popraw dane"
    :stop
125 key 1,"sys"+str$(pocz)+chr$(13)
130 printchr$(148)"uruchomienie programu
    : klawisz f1"
135 :
140 rem ***** kod maszynowy *****
145 :
150 data 032,027,227,032,141,227,032,211
155 data 232,032,216,251,013,013,078,065
160 data 090,087,065,058,032,034,000,162
165 data 000,189,055,003,032,076,255,232
170 data 224,017,208,245,032,216,251,034
175 data 013,013,080,082,079,071,082,065
180 data 077,032,087,032,000,173,121,003
185 data 201,136,240,042,201,134,240,058
190 data 201,025,240,082,201,190,240,089
195 data 032,216,251,013,083,089,083,084
200 data 069,077,032,078,073,069,032,082
205 data 079,090,080,079,090,078,065,078
210 data 089,046,013,013,000,096,032,216
215 data 251,084,085,082,066,079,032,084
220 data 087,073,068,068,089,046,013,013
225 data 000,096,032,216,251,072,083,032
230 data 084,085,082,066,079,032,040,075
235 data 082,065,075,079,087,073,065,075
240 data 041,046,013,013,000,096,032,216
245 data 251,084,083,076,046,013,013,000
250 data 096,032,216,251,078,079,086,065
255 data 076,079,065,068,013,013,000,096
```

PAMIĘTNIK ARTYLERZYSTY — odc. IV

UWAGA:
Do wpisania programów zamieszczonych w tym artykule NIEZBĘDNY będzie program typu MONITOR lub ASEMBLER.

Tym razem nadszedł czas, by przedstawić poważnie kilka najważniejszych zasad i pojęć związanych z rastrem. Być może start był zbyt ostry, więc nie każdy zdążył się zorientować, co się dzieje. Przypomnijmy więc:

1. Obraz na ekranie monitora jest tworzony przez wiązkę elektronów odpowiednio kierowaną przez działo zainstalowane w kineskopie. Im większe jest natężenie wiązki, czyli im więcej elektronów wyrzucamy w stronę ekranu, tym obraz jest jaśniejszy. Wynika z tego, że ekran nie jest tworzony od razu cały: wiązka jest bowiem jedynie jednym punktem na tym obrazie. Jak się więc dzieje, że zmiany zachodzą na całym ekranie, a nie tylko w jednym punkcie? Problem ten wynalazcy telewizji (bo monitor — to po prostu szczególny rodzaj telewizora, tyle tylko, że obraz odbiera ze stojącego przed nim komputera, a nie z odległej stacji przekaźnikowej) rozwiązali w prosty sposób — pokryli wewnętrzną powierzchnię kineskopu cienką warstwą substancji zwaną luminoforem, która ma w zwyczaju świecić, gdy trafi w nią wiązka elektronów. Co ciekawe świeci jeszcze przez chwilę gdy przestaniemy w nią „strzelać" tymi elektronami. Dzielni wynalazcy zmusili teraz działo, by strzelało w coraz to inny punkt. Dzięki temu kontrolować możemy bez problemu cały ekran.
2. Aby wytworzyć obraz, działo celuje najpierw w lewy górny róg obrazu. Wstrzeliwuje tam odpowiednią liczbę elektronów i zaczyna celować o dokładnie jeden punkt w prawo. Wyświetla tam punkt numer 2 — drugi punkt od lewej w górnej linii. W ten sposób, krok po kroku, wyświetla całą najwyższą linię ekranu. Kiedy dojdzie już do końca, zawraca znowu do lewego krańca, przeskakując jednocześnie do drugiej linii (dokładnie tak, jak maszyna do pisania). W drugiej linii cały proces się powtarza. Kiedy już, wyświetliwszy cały ekran, dojdzie do dolnego — prawego rogu, błyskawicznie zawraca do rogu górnego — lewego. Trwa to dokładnie 0.02 (jedną pięćdziesiątą) sekundy.

To właśnie powoduje, że programiści (albo, jeśli wolicie — koderzy) zajęli się rastrem. W naszym cyklu omawialiśmy już zagadnienie w jaki sposób można zmieniać np. kolor ekranu, czy jego przesunięcie w poziomie w każdej linii. Dzięki temu powstają czasami wręcz niezwykłe efekty.

Skąd właściwie one się biorą? Dobre pytanie. Wyświetlenie przez kineskop całego ekranu trwa 1/50 sekundy. Dla komputera, nawet tak powolnego jak C-64 1/50 sekundy to, wierzcie mi, kawał czasu. Dlatego też, jeśli odpowiednio zsynchronizujemy działania naszej armaty i starego, dobrego komputera, możemy oczekiwać dobrych efektów. Jeśli każemy procesorowi wykonać jakikolwiek rozkaz, choćby i najprostszy — NOP, zajmiemy mu troszkę czasu. W tym samym czasie działo zmieni odrobinę punkt, w którym będzie celować. Dlatego też, jeżeli po tym rozkazie każemy jeszcze zmienić kolor ekranu, nie dziwmy się, że kolor zmieni się w miejscu trochę bardziej oddalonym w prawo od lewego marginesu. Abyś nie musiał wierzyć mi na słowo, wpisz programik oznaczony jako LISTING 1.

Uruchom go przez G2710 lub SYS 10000. W górnej części ekranu, z lewej strony, zobaczysz szarpiący się kawałek jaśniejszej linii. Mało efektowne, za to dydaktyczne i posłuży nam głównie do sprawdzenia tego, co napisałem wcześniej. Między INC $d021 a DEC $d021 dopisz jeszcze kilka rozkazów NOP. I co? Zgadza się? Tak! Linijka robi się tym dłuższa, im więcej razy dopiszesz rozkazów NOP. Przy kilkudziesięciu możemy doprowadzić do tego, że cała linia będzie jaśniejsza. Mam nadzieję, że to już ostatecznie tłumaczy, skąd i dlaczego czasem pojawiają się szarpania kolorów, gdy chcemy je zmieniać w poszczególnych liniach. Wiemy też, jak im zapobiegać, albo je likwidować — należy dodać tak wiele instrukcji NOP, by migotanie weszło pod obwódkę, gdzie będzie niegroźne i niewidoczne.

Takie właśnie chytre uniki nazywa się „docyklowaniem". „Cyklować" można również innymi instrukcjami: dopuszczalne są np. pary PHA — PLA, rozkazy BIT $00 itp. Generalnie rzecz biorąc polecenia te nie powinny mieć wpływu na pracę programu.

W poprzednich odcinkach „Pamiętnika", przedstawiłem również metody „cyklowania" za pomocą pętli. Można, np. za pomocą JSR skakać do odpowiedniego punktu w tabeli, zawierającej wyłącznie wartości $ea (kod rozkazu NOP). W disasemblerze wyglądałaby ona tak:

```
NOP
NOP
NOP
.
.
.
NOP
NOP
RTS
```

Teraz — zależnie od tego, jakie chcesz uzyskać opóźnienie, tak dalece przed rozkazem RTS musisz wskoczyć. Pod pozorem „docyklowania" możemy też nasz program zabezpieczyć przed ciekawskimi, a niedouczonymi. Bardzo spodobał mi się mały patent, który podpatrzyłem u pana Polonusa w jego małym demku „Amiga Scroll". Jest gdzieś w nim rozkaz STA, $cfff,X. Z pozoru rozkaz ten nie ma żadnego głębszego sensu. Sens jednak jest, ale głęboko ukryty: daleko przed tym rozkazem stoi sobie LDX #$19. Dzięki temu STA $cfff,X trafia dokładnie tam gdzie trzeba — do $d018. Przy okazji — wykonanie takiego rozkazu trwa jeden cykl dłużej, niż gdyby był w trybie bezpośrednim. Pan Polonus upiekł tu dwie pieczenie na jednym ogniu — zabezpieczył swój program przed ciekawskimi i dokładnie go „docyklował".

Oczywiście, „cyklowanie" w poziomie to nie wszystko. Jest jeszcze równie ważne, jeśli nie ważniejsze — „cyklowanie" w pionie. Jeżeli bowiem wywołamy przerwanie na samej górze ekranu, wykonamy kilka operacji i będziemy chcieli zejść kilka czy kilkanaście linii niżej, to co? Czy będziemy stosować jakieś makabryczne ilości NOPów? A może pętle opóźniające? Nie. Zrobimy to najprościej i najskuteczniej, jak mogę sobie wyobrazić. (Uwaga Czytelnicy! Jeśli ktoś zna jakiś szybszy, prostszy i skuteczniejszy sposób, niech koniecznie się nim z nami podzieli!) Kiedy już w poprzedniej linii zrobimy wszystko, co trzeba, ładujemy do akumulatora numer linii, do którego chcemy dotrzeć. Porównujemy go z zawartością rejestru RASTER ($d012). Jeżeli się nie zgadza, porównujemy jeszcze raz. Komórka RASTER ciągle zmienia swą zawartość, więc po jakimś czasie NA PEWNO trafimy tam, gdzie chcieliśmy. To tak, jak z windą, która zatrzymuje się na każdym piętrze. Kiedy zatrzyma się na naszym — wysiadamy (czyli przestajemy powtarzać pętlę w nieskończoność). Program taki przedstawia LISTING 2.

Jest to procedura dość prosta: W liniach $2710 — $2726 zmieniamy wektor przerwań IRQ i uzależniamy go od rastra. Następnie ($2729 — $273f) kilkakrotnie zmieniamy kolor tła. W liniach $2742 — $2747 tkwi samo sedno sprawy. To tutaj siedzi właśnie ta pętla, która ma za zadanie odczekać do zadanej linii ekranu. Potem już tylko zmieniamy kolor tła na poprzednie i wychodzimy z przerwania.

Proszę zwrócić uwagę na to, że w linii o numerze $50 kilkakrotnie zmieniamy kolor ekranu. Taki efekt nazywany jest SPLITEM, czyli — po polsku — rozcięciem. Rozcinamy tu ekran zmieniając jego kolor w jednej linii kilkakrotnie. Oczywiście, ten przykład jest skrajnie prymitywny, ale w jednym z przyszłych numerów pokażę, jak można robić ładne, równe splity.

A na razie — poćwiczcie! Poeksperymentujcie!

BARTEK I. KACHNIARZ

KULKI

Zamieszczony poniżej programik jest miłym dla oka zobrazowaniem złożenia dwóch funkcji sinus. Poza tym procedury o podobnym działaniu są stosowane w bardzo wielu programach demonstracyjnych przeznaczonych na C-64. Warto zaznaczyć, że „Kulki" działają w przerwaniach IRQ, a widoczne na ekranie latające obiekty to po prostu duszki (sprite'y).

Program ma poza tym wbudowany edytor, służący do zmiany trajektorii ruchu widocznych na ekranie obiektów.

LISTING 1

```
A 2710 SEI
A 2711 LDA #$40
A 2713 CMP $d012
A 2716 BNE $2713
A 2718 INC $d021
A 271b NOP
A 271c NOP
A 271d DEC $d021
A 2720 CMP $d012
A 2723 BEQ $2720
A 2725 BNE $2711
```

LISTING 2

```
A2710 SEI
A2711 LDA #$27
A2713 STA $0315
A2716 LDA #$29
A2718 STA $0314
A271b LDA #$01
A271d STA $d01a
A2720 LDA #$7f
A2722 STA $dc0d
A2725 CLI
A2726 JMP $2726
A2729 LDA $d012
A272c CMP $d012
A272f BEQ $272c
A2731 STA $d021
A2734 INC $d021
A2737 INC $d021
A273a INC $d021
A273d LDA #$00
A273f STA $d021
A2742 LDA #$60
A2744 CMP $d012
A2747 BNE $2744
A2749 LDA #$06
A274b STA $d021
A274e LDA #$50
A2750 STA $d012
A2753 LDA #$1b
A2755 STA $d011
A2758 INC $d019
A275b JMP $ea81
```

```
98 rem * kulki *
99 :
100 a=50176:p=180/3.1415926:prin
    t"obliczam tablice funkcji t
    rygonometr."
105 for t=0 to 359 step 360/256:
    poke a,64+64*cos(t/p)
110 poke a+256,44+44*cos(t/p):a=
    a+1:next
115 gosub 230
120 :
125 poke 896,0:poke 897,0:poke 8
    98,0:poke 899,0
130 poke 900,21:poke 901,21:poke
    902,21:poke 903,21
135 poke 904,1:poke 905,2:poke 9
    06,2:poke 907,3
140 sys 49152:for t=1 to 8:a(t)=
    899+t:next
145 :
150 print chr$(5);chr$(147)
155 print:print:nr=0
160 print"1.parametr x1=";peek(9
    00)
165 print"2.parametr x2=";peek(9
    01)
170 print"3.parametr y1=";peek(9
    02)
175 print"4.parametr y2=";peek(9
    03)
180 print"5.predkosc x1=";peek(9
    04)
185 print"6.predkosc x2=";peek(9
    05)
190 print"7.predkosc y1=";peek(9
    06)
195 print"8.predkosc y2=";peek(9
    07)
200 print:print"podaj numer para
    metru do zmiany:"
205 input"numer (1-8):";nr
210 if nr<1 or nr>8 then 145
215 print:print"aktualna wartosc
    :";peek(a(nr))
220 input"nowa wartosc:";w
225 poke a(nr),w and 255:goto145
230 print:print"jeszcze kod masz
    ynowy..."
235 restore:c=o:d=49152
240 read a$:if a$="end" then 275
245 n1=asc(left$(a$,1))and63:n2=
    asc(right$(a$,1))and63
250 if n1<48 then n1=n1+9:goto 2
    60
255 n1=n1-48
260 if n2<48 then n2=n2+9:goto 2
    70
265 n2=n2-48
270 a=n1*16+n2:poke d,a:c=c+a:d=
    d+1:goto 240
275 if c<>29778 then print"zle d
    ane!":stop
280 return
285 data 78,a9,7f,8d,0d,dc,a2,00
290 data 8e,0e,dc,e8,8e,1a,d0,a9
295 data 1b,8d,11,d0,a9,ff,8d,12
300 data d0,a2,3f,bd,c2,c0,9d,40
305 data 03,ca,10,f7,8e,15,d0,8e
310 data 1c,d0,8e,1b,d0,8d,17,d0
315 data 8d,1d,d0,a2,07,a9,0d,9d
320 data f8,07,a9,01,9d,27,d0,ca
325 data 10,f3,a9,00,8d,20,d0,8d
330 data 21,d0,a9,02,8d,25,d0,a9
335 data 0a,8d,26,d0,a9,60,8d,14
340 data 03,a9,c0,8d,15,03,58,60
345 data 20,69,c0,ee,19,d0,4c,31
350 data ea,a2,00,bd,80,03,48,e8
355 data e0,04,d0,f7,a2,00,ac,80
360 data 03,b9,00,c4,ac,81,03,18
365 data 79,00,c4,18,69,2c,6e,10
370 data d0,9d,00,d0,ac,82,03,b9
375 data 00,c5,ac,83,03,18,79,00
380 data c5,18,69,36,9d,01,d0,a0
385 data 03,b9,80,03,18,79,84,03
390 data 99,80,03,88,10,f3,e8,e8
395 data e0,10,d0,c2,a2,03,68,18
400 data 7d,88,03,9d,80,03,ca,10
405 data f5,60,00,55,00,01,bb,40
410 data 07,af,d0,06,bb,d0,1e,af
415 data d4,1a,bf,74,7a,ef,d4,7a
420 data ad,d5,6a,bf,5d,6a,ed,d5
425 data 6a,bf,55,6a,ff,75,6a,bd
430 data d5,1b,bf,54,1e,fd,d4,1b
435 data fd,54,05,f7,50,07,dd,50
440 data 01,55,40,00,55,00,00,00
445 data 00,00,00,00,00,00,end
```

PCHEŁKA

Pchełki to takie programy, które przy minimum objętości dają duże efekty. Myślę tu o procedurze nazwanej tu LISTING 1. Program ten należy wpisać za pomocą dowolnego makroasemblera (np. TURBOASS V5.0) i uruchomić. Ciekawy efekt, nieprawdaż?

Cały trick polega na manipulacji zawartością komórki $d011, a dokładniej czterema młodszymi bitami odpowiadającymi za pionowy przesuw ekranu oraz za szerokość górnego i dolnego marginesu. Ciągła zmiana zawartości tego rejestru wprowadza układ VIC w błąd, co pociąga za sobą nieprzewidziane skutki.

Poeksperymentujmy jeszcze z tym rejestrem. Wpisz i uruchom program opisany jako listing 2.

Wciskając klawisz RESTORE zmieniasz efekty. Zabawa tym rejestrem, przy odrobinie wprawy, może zaowocować niezwykłymi rozwiązaniami. Wiele takich efektów w programach demonstracyjnych to nic innego jak sprytna manipulacja rejestrem $d011. Dzięki niemu można np. uzyskać przesuw zawartości ekranu w płaszczyźnie poziomej (!), bez dokonywania zmian w pamięci obrazu. Życzę udanej zabawy i ciekawych eksperymentów z tym rejestrem.

JETBOY

```
        ;LISTING 1
        ;
        *= $c000
        SEI
        LDA #$11
        STA $3fff
loop    LDA $d012
        ADC #$06
        AND #$1f
        STA $d011
        NOP
        STA $d020
        JMP loop
```

```
        ;LISTING 2
        ;
        *= $c000
        LDA #$11
        STA $3fff
        SEI
        LDX #<restore
        LDY #>restore
        STX $0318
        STY $0319
loop    LDA $d012
        ADC #$00
        AND #$1f
        STA $d011
        NOP
        STA $d020
        JMP loop
restore INC loop+4
        RTI
```

LUPA dla C-16/116/PLUS4/C-128

Historia tego programu jest następująca: zobaczyłem program, w którego czołówce pojawiały się „duże hasła". Postanowiłem napisać programik (w BASIC V3.5) powiększający dowolny obszar ekranu w trybie wysokiej rozdzielczości.

Tym właśnie programem jest zamieszczona obok „LUPA". „LUPA" działa bardzo sprytnie: każdemu punktowi graficznemu przyporządkowuje prostokąt (linia 650) o odpowiednich współrzędnych. Prostokąt może być zastąpiony przez kółko, trójkąt lub dowolny wielokąt (dzięki zastosowaniu instrukcji CIRCLE zamiast BOX — patrz wiersz 510).

Jak się za „to" wziąć? Najpierw należy wpisać program do komputera i zapisz przed pierwszym uruchomieniem. Teraz wciśnij przycisk RESET. Po pojawieniu się czołówki wykonaj GRAPHIC 2,1 i przykładowo CHAR 1,0,0,"#". W górnym lewym rogu ekranu powinien pokazać się krzyżyk. Teraz wczytaj i uruchom „Lupę". Proponuję wpisać kolejno następujące dane: x1=0, y1=0, x2=7, y2=7, 16, 16, 100, 40, K, N. Na środku ekranu powinien narysować się duży krzyż. Aby dobrze posługiwać się „Lupą", trzeba poświęcić trochę czasu na eksperymenty. Powodzenia!

KUBA MAŁECKI

```
100 rem *************************
105 rem *  lupa dla commodore   *
110 rem *    c-16/116/+4/128    *
115 rem *                       *
120 rem *     kuba malecki      *
125 rem *************************
150 :
155 graphic 0,1
160 char 1,1,1,"podaj wspolrzedne
    obszaru powiekszanego"
165 char 1,1,3,"0<= x1 < x2 < 320"
170 char 1,1,4,"0<= y1 < y2 < 200"
175 char 1,1,6,"wspolrzedna x1= "
180 input x1
185 if x1<0 or x1>319 then 175
190 char 1,1,7,"wspolrzedna y1= "
195 input y1
200 if y1<0 or y1>199 then 190
205 char 1,1,8,"wspolrzedna x2= "
210 input x2
215 if x2<=x1 or x2>319 then 205
220 char 1,1,9,"wspolrzedna y2= "
225 input y2
230 if y2<=y1 or y2>199 then 220
235 char 1,1,11,"dobre dane (t/n)"
240 input a$
245 if a$<>"n" and a$<>"t" then 2
    35
250 if a$="n" then 175
255 :
260 scnclr
265 char 1,1,1,"iloczyn pow. w ki
    erunku x = ":input szer
270 if szer<=1 then 265
275 char 1,1,3,"iloczyn pow. w ki
    erunku y = ":input wys
280 if wys<1 then 275
285 char 1,1,5,"wspolrzedna x1= "
290 input xx1
295 if xx1<0 or xx1>319 then 285
300 char 1,1,6,"wspolrzedna y1= "
305 input yy1
310 if yy1<0 or yy1>199 then 300
315 char 1,1,8,"dobre dane (t/n)"
320 input z$
325 if z$<>"n" and z$<>"t" then 3
    15
330 if z$="n" then 265
335 :
340 scnclr:print
345 input" kwadraty czy wielokaty
    (k/w) ";kw$:print
350 if kw$<>"k" and kw$<>"w" then
    370
355 if kw$="w" then 375
360 input" normalnie czy inwersja
    (n/i) ";inw$
365 if inw$<>"n" and inw$<>"i" th
    en 340
370 goto 390
375 input" ilosc bokow  (3,4,...,
    360) ";boki
380 if boki<3 or boki>360 then 375
385 :
390 graphic 1
395 x0=(xx1/szer):y0=yy1
400 if inw$="i" then 445
405 for y=y1 to y2
410 :for x=x1 to x2
415 ::locate x,y
420 ::if rdot(2)=1 then gosub 485
425 :next x
430 next y
435 goto 525
440 :
445 for y=y1 to y2
450 :for x=x1 to x2
455 ::locate x,y
460 ::if rdot(2)=0 then gosub 485
465 :next x
470 next y
475 goto 525
480 :
485 xz=x-x1:yz=y-y1
490 kx=(x0+xz)*szer:ky=yz*wys+y0
495 if kw$="w" then 510
500 box 1,kx,ky,kx+szer-1,ky+wys-
    1,,1
505 return
510 circle 1,kx+szer,ky+wys,szer/
    2,wys/2,,,,360/boki
515 goto 505
520 :
525 getkey a$:graphic 0,1
```

JOYSTICK i DUSZKI

Ostatnio do redakcji przychodzi sporo listów z zapytaniem, w jaki sposób można poruszać duszkami za pomocą joysticka. Ponieważ temat ten pojawiał się często w korespondencji adresowanej także do Bajtka, postanowiłem opublikować po prostu odpowiedni program. Procedurka ta jest bardzo prosta, tym niemniej podaję jej objaśnienie aby rozwiać wszelkie wątpliwości początkujących — wszak to oni właśnie życzyli sobie takiego programu.

W linii 100 jest czyszczony ekran. Następnie (wiersz 105) przypisuje zmiennej X współrzędną poziomą duszka nr 0, zmiennej Y — współrzędną pionową, a zmiennej SB — adres rejestru układu VIC. Po wpisaniu do tego rejestru odpowiedniej wartości duszek może przemieszczać się aż do prawej ramki. Jak wiadomo, do komórki pamięci można wpisać co najwyżej liczbę 255, a przecież rozdzielczość ekranu wynosi 320 punktów (poziomo). Zatem, aby duszek mógł poruszać się w prawo bez ograniczeń, do komórki o adresie 53264 należy wpisać 1 (patrz wiersz 215). Dotyczy to duszka nr 0; dla innych duszków należy wpisywać wartości będące kolejnymi potęgami dwójki.

Wiersz 110 ustawia duszka na środku ekranu. Linie 115 i 120 określają kształt duszka — w naszym programie jest to zwykły, wypełniony prostokąt. Instrukcja POKE X+21,1 aktywizuje (włącza) naszego duszka. Linie 125-150 odczytują stan joysticka, a pozostałe powodują odpowiedni do tego stanu ruch duszka. Ot, i cała filozofia.

Program działa niestety dość wolno, ale nie należy oczekiwać zbyt wiele od programu napisanego w BASIC. Pamiętaj również, że dodatkowo są tu jeszcze sprawdzane pewne warunki przed każdym poruszeniem duszka o jeden punkt — w przeciwnym wypadku duszek „chowałby się" poza ramką. Działanie procedurki można przyspieszyć, wpisując zamiast jedynek w liniach 165, 180, 200 i 220 wartość np. 2, z tym że wtedy należy też zmodyfikować odpowiednie wartości ograniczające ruch duszka poza ramkę. Tu musisz już sam poeksperymentować.

Jeśli dla kogoś szybkość działania tego programu jest zbyt mała, to proponuję zajrzeć do C&A nr 03/92, gdzie w artykule „Jak napisać własną grę" podałem listing podobnego programu napisanego w języku maszynowym.

```
100 print chr$(147)
105 x=53248:y=53249:sb=53264
110 poke x,170:poke y,132
115 for i=832 to 896:poke i,255:
    next
120 poke 2040,13:poke x+21,1
125 joy=peek(56320)
130 if (joyand1)=0 then 160
135 if (joyand2)=0 then 175
140 if (joyand4)=0 then 190
145 if (joyand8)=0 then 210
150 goto 125
151:
155 rem === sprite do gory ===
156 :
160 if peek(y)=50 then 125
165 poke y,peek(y)-1:goto 125
169 :
170 rem === sprite w dol ===
171 :
175 if peek(y)=230 then 125
180 poke y,peek(y)+1:goto 125
184 :
185 rem === sprite w lewo ===
186 :
190 if peek(x)=0 and peek(sb)=1
    then poke x,255:poke sb,0:g
    oto 125
195 if peek(x)=24 and peek(sb)=0
    then 125
200 poke x,peek(x)-1:goto 125
204 :
205 rem === sprite w prawo ===
206 :
210 if peek(sb)=1 and peek(x)>63
    then 125
215 if peek(x)=255 then poke sb,
    1:poke x,0:goto 125
220 poke x,peek(x)+1:goto 125
```

CHRISTIAN GRZENKOWICZ

ASEMBLER kontra BASIC

Zapewne słyszeliście nieraz, że język maszynowy jest szybszy od BASIC. Ci bardziej zaawansowani wiedzą dlaczego tak się dzieje — aby wykonać jedną, najprostszą instrukcję BASIC, komputer musi wykonać przynajmniej kilkadziesiąt rozkazów asemblera. Program zamieniający instrukcje języka BASIC na rozkazy asemblera nazywa się interpreterem BASIC.

Szybkość wykonywania programów w asemblerze bierze się z tego, że procesor może wykonywać je bezpośrednio — gotowy program w języku maszynowym trzeba tylko uruchomić instrukcją SYS. Mimo to, asembler ma licznych przeciwników, którzy twierdzą, że szybkość wykonywania programu w asemblerze jest rekompensowana w BASIC łatwością obsługi. To nieprawda. Asembler, istotnie, jest prymitywny — ma kilkadziesiąt prościutkich rozkazów, ale — jeśli program wykonywany jest często i zawiera dużo obliczeń na liczbach całkowitych — pisanie programów w tym właśnie języku stokrotnie się opłaca.

Dowodem tego są dwa krótkie programiki, zamieszczone w tym artykule. Najpierw wpisz pierwszy z nich. Łatwo się domyślić, co ma zrobić — doliczyć do miliona. No, dobrze, niech liczy. W tym czasie radzę ci zjeść obiad, odrobić lekcje, wyjść na spacer z dziewczyną, pójść na dyskotekę, obejrzeć kino nocne... No, dość, powinno już starczyć. Program działał z górą pięć i pół godziny.

A teraz wpisz program drugi. Nie martw się jego zewnętrzną formą. Ten BASIC służy tylko do wpisania do pamięci programu w asemblerze. Tak po prostu najwygodniej było to przedstawić w naszym piśmie. Teraz napisz RUN, doczekaj, aż sekundnik w Twoim zegarku dojdzie do dwunastki i dopiero teraz naciśnij RETURN. Wzrokiem można śledzić co najwyżej dziesiątki tysięcy — tysiące skaczą tak szybko, że najmocniejsi wysiadają. Cały program powinien się wykonać w ciągu mniej więcej 25 (słownie: dwudziestu pięciu) sekund!

No i proszę. Program w asemblerze wykonywał się jakieś 1000 razy szybciej. A teraz — musisz jeszcze tylko nauczyć się asemblera! Zwłaszcza, że w C&A właśnie zaczynamy kurs tego prymitywnego, acz zaskakująco skutecznego języka.

BARTEK I. KACHNIARZ

LISTING 1

```
10 FOR A=1 TO 1000000
20 PRINT CHR$(19);A
30 NEXT
```

LISTING 2

```
101 rem ** assembler vs basic **
102 rem **      c&a 1992      **
103 rem **  bartek kachniarz  **
105 :
109 read a
110 if a<0 then 150
120 poke 10000+b,a
130 b=b+1:c=c+a
140 goto 100
150 if c=4363 then sys 10000:stop
160 print"niedobre dane!"
161 :
200 data 169,048,162,000,157,001,
    004
210 data 232,224,006,208,248,254,
    001
220 data 004,189,001,004,201,058,
    208
230 data 012,169,048,157,001,004,
    202
240 data 208,238,238,001,004,096,
    162
250 data 006,208,230,-1
```

64 DUSZKI na EKRANIE

Jak zapewne wiesz, Commodore 64 ma możliwość wyświetlania tzw. „duszków" czyli niezależnych, ruchomych obiektów graficznych. W instrukcji obsługi jest napisane, że można ich wyświetlić na ekranie maksymalnie osiem na raz. Z kolei w niejednej grze widziałeś mnóstwo duszków poruszających się po ekranie i na pewno było ich tam więcej niż osiem; kto więc ma rację?

Otóż tak naprawdę Commodore potrafi pokazać jednocześnie tylko osiem duszków. To, że widzimy ich więcej jest spowodowane różną interpretacją obrazu przez komputer i ludzki mózg. Człowiek patrząc na obraz widzi go od razu w całości. Komputer generując obraz robi to trochę inaczej. Rysunek na ekranie monitora jest tworzony przez wiązkę elektronów zwaną rastrem. Gdy przyjrzysz się z bliska ekranowi, zauważysz cieniutkie poziome linie; są to właśnie ślady przebiegu tej wiązki. Dla komputera obraz składa się z serii punktów ułożonych jeden za drugim. Komputer w danym, bardzo krótkim momencie (ok. 125*10E-9 sekundy) wyświetla tylko jeden punkt. Dla naszych rozważań przyjmijmy, że komputer wyświetla naraz jedną linię rastra. W jednej takiej linii może być tylko osiem duszków.

Spróbujmy oszukać komputer; ustaw mu wszystkie osiem duszków tak, aby ich wyświetlanie rozpoczęło się w linii rastra o numerze x, (trzeba zauważyć, że pozycja Y duszka odpowiada numerowi rastra, od którego zaczyna się jego wyświetlanie). Komputer wyświetli więc osiem duszków. W momencie, gdy wyświetlana jest linia rastra numer x+22, duszki nie są już używane. Wystarczy więc przesunąć je do linii rastra x+23, i znowu pojawią się one na ekranie. Zauważ, że naraz wyświetlanych jest osiem duszków, natomiast my widzimy ich szesnaście, ponieważ oko ludzkie charakteryzuje się dużą bezwładnością. Ta technika nazywana jest Sprite-multiplexer i jest bardzo często używana w grach oraz demach.

A teraz przykładowy program, który wyświetla na ekranie 64 duszki. Wciskając kombinację klawiszy „Control" + „Commodore logo" zmieniasz priorytet duszków, to znaczy ustalasz czy duszki mają być wyświetlane nad czy pod tekstem.

```
100 rem *********************
105 rem *    64  duszki     *
110 rem * by jetboy/parados *
115 rem *********************
116 :
120 d=16320:b=5844
125 gosub 150
130 d=49152:b=52286
135 gosub 150
140 sys e
145 end
150 c=0:e=d
155 read a$:if a$="end" then 205
160 a1=asc(left$(a$,1))and 63
165 a2=asc(right$(a$,1))and 63
170 if a1>47 then 180
175 a1=a1+9:goto 185
180 a1=a1-48
185 if a2>47 then a2=a2-48:goto
    195
190 a2=a2+9
195 a=a1*16+a2:poke d,a
200 d=d+1:c=c+a:goto 155
205 if c<>b then print"blad w li
    niach data":stop
210 return
215 data 00,00,00,00,00,00,00,00
220 data 00,07,80,3f,1f,80,7f,3f
225 data 80,7f,78,00,f7,70,40,e7
230 data e0,a1,e7,e0,c1,c7,e1,23
235 data c7,e0,d3,ff,e0,07,ff,70
240 data 07,ff,78,0f,07,3f,ff,07
245 data 1f,fe,07,07,fe,07,00,00
250 data 00,00,00,00,00,00,00,00
    ,end
255 data 78,20,5b,ff,a2,c0,a0,50
260 data 8c,14,03,8e,15,03,a9,01
265 data 8d,1a,d0,a9,7f,8d,0d,dc
270 data a9,30,8d,12,d0,a9,1b,8d
275 data 11,d0,a9,ff,8d,15,d0,a9
280 data 00,aa,9d,27,d0,e8,e0,08
285 data d0,f8,a2,00,a9,39,9d,00
290 data d0,18,69,20,e8,e8,e0,10
295 data d0,f4,a2,00,a9,ff,9d,f8
300 data 07,e8,e0,08,d0,f8,58,60
305 data ee,19,d0,ce,19,d0,a9,34
310 data a2,00,9d,01,d0,e8,e8,e0
315 data 10,d0,f7,a2,c0,a0,75,8c
320 data 14,03,8e,15,03,a9,49,8d
325 data 12,d0,4c,81,ea,ee,19,d0
330 data ce,19,d0,a9,4d,a2,00,9d
335 data 01,d0,e8,e8,e0,10,d0,f7
340 data a2,c0,a0,9a,8c,14,03,8e
345 data 15,03,a9,63,8d,12,d0,4c
350 data 81,ea,ee,19,d0,ce,19,d0
355 data a9,66,a2,00,9d,01,d0,e8
360 data e8,e0,10,d0,f7,a2,c0,a0
365 data bf,8c,14,03,8e,15,03,a9
370 data 7c,8d,12,d0,4c,81,ea,ee
375 data 19,d0,ce,19,d0,a9,7f,a2
380 data 00,9d,01,d0,e8,e8,e0,10
385 data d0,f7,a2,c0,a0,e4,8c,14
390 data 03,8e,15,03,a9,95,8d,12
395 data d0,4c,81,ea,ee,19,d0,ce
400 data 19,d0,a9,98,a2,00,9d,01
405 data d0,e8,e8,e0,10,d0,f7,a2
410 data c1,a0,09,8c,14,03,8e,15
415 data 03,a9,ae,8d,12,d0,4c,81
420 data ea,ee,19,d0,ce,19,d0,a9
425 data b1,a2,00,9d,01,d0,e8,e8
430 data e0,10,d0,f7,a2,c1,a0,2e
435 data 8c,14,03,8e,15,03,a9,c7
440 data 8d,12,d0,4c,81,ea,ee,19
445 data d0,ce,19,d0,a9,ca,a2,00
450 data 9d,01,d0,e8,e8,e0,10,d0
455 data f7,a2,c1,a0,53,8c,14,03
460 data 8e,15,03,a9,e0,8d,12,d0
465 data 4c,81,ea,ee,19,d0,ce,19
470 data d0,a9,e3,a2,00,9d,01,d0
475 data e8,e8,e0,10,d0,f7,a2,c0
480 data a0,50,8c,14,03,8e,15,03
485 data a9,30,8d,12,d0,ae,01,dc
490 data e0,db,d0,08,ad,1b,d0,49
495 data ff,8d,1b,d0,4c,31,ea,00
    ,end
```

RAFAŁ PIASEK

BEEP 64

Kiedyś, gdy byłem jeszcze w epoce BASIC narzekałem na trudności w stworzeniu jakiejkolwiek muzyczki. Stosowanie POKE przyprawiało mnie o zawrót głowy. Rozwiązałem problem „głuchoty" programów za pomocą jednego dźwięku, który raz uruchomiony jest „ciągle żywy". Programik (38 bajtów) napisany jest w asemblerze, jego „instalację" przedstawiam w BASIC.

Program znajduje się w obszarze pamięci CO50-CO75, dzięki czemu nie koliduje z programem napisanym w języku BASIC. Podaję także jak można wyłączyć dźwięk oraz ponownie go włączyć. Wykonywanie jakichkolwiek czynności przez program wywołujący dźwięk nie zagraża zablokowaniem komputera.

Wyłączenie dźwięku: POKE 49236, 0 : SYS 49232

Włączenie dźwięku : POKE, 49236, 192 : SYS 49232

SZYMON WERTYŃSKI

```
100 REM AUTOR: SZYMON WERTYNSKI
120 REM
125 :
130 FOR X=49232 TO 49269:READ Y:POKE
X,Y : NEXT
135 SYS 49232
140 DATA 162,026,189,091,192,157,255,211
145 DATA 202,208,247,096,048,004,000,000
150 DATA 033,015,250,040,004,000,000,033
155 DATA 015,250,024,002,000,000,033,015
160 DATA 250,000,016,255,063,065
```

[...] Mam poważny problem z wczytaniem i uruchomieniem programu GEOS V1.3, który otrzymałem kupując myszkę (MOUSE 1351). Cały program jest zbiorem USR. [...]

Dyskietka sprzedawana w komplecie z myszką 1351 zawiera na swojej pierwszej stronie programy służące do jej uruchomienia, natomiast na drugiej stronie tzw. GEOS UPGRADE V1.3 (lub V1.5). Generalnie rzecz biorąc plik ten pozwala na zamianę starszej wersji systemu na nowszą. Oprócz więc tego pliku musisz mieć jeszcze oryginalną dyskietkę systemową z systemem GEOS niższym od V1.3 (lub V1.5).

Czy jest możliwość przyspieszenia wczytywania programów z magnetofonu? Czy wystarczy tylko przyspieszyć obroty silnika? [...]

Aby przyspieszyć wczytywanie się programów z magnetofonu można użyć programu zwanego popularnie TURBO (np. Turbo ROM, ABC, SuperTurbo itp.). Zadaniem tego programu jest przyspieszenie transmisji komputer-magnetofon, co odbywa się w sposób programowy. Sprawy nie rozwiąże absolutnie przyspieszenie obrotów silnika.

[...] Mam moduł FINAL III i kilka kłopotów z nim związanych:
1. Gdy chcę wczytać grę zapisaną z opcji Freeze na kasecie, wgrywam pierwszą część gry o nazwie FC i wszystko jest w porządku do czasu, gdy chcę wczytać drugą część o nazwie —FC. Połowa sygnału nagłówka jest zmazana przez poprzednią część FC. Dlaczego?
2. Gdy chcę przegrać program z poziomu języka BASIC, nie mogę zapisać go z żadną nazwą. Gdy zapiszę nazwę i zatwierdzę komunikat zapisu, komputer wyświetla mi „OUT OF MEMORY ERROR". Dlaczego?
3. Gdy wchodzę w opcję Preferences i chcę zmienić numer urządzenia, z którego aktualnie korzystam (magnetofon), nie mogę przestawić wskaźnika na 1, mogę natomiast zmienić numery na 8 i 9. Dlaczego?

1. Program z opcji FREEZE zapisywany jest w postaci dwóch plików o nazwach FC i —FC. Pierwszy z nich (FC) zawsze wczytuje drugi (—FC). Aby wczytać program zapisany w ten sposób należy wczytać plik FC i go uruchomić (nie wyłączając magnetofonu). Po wpisaniu RUN i wciśnięciu RETURN komputer automatycznie doczyta drugą część programu i uruchomi całość.
2. Przy dłuższych programach podczas próby zapisu ich na dyskietce lub kasecie pojawia się komunikat OUT OF MEMORY ERROR. Jest to spowodowane długością programu — interpreter BASIC ma zbyt mało miejsca na zapisanie tytułu w pamięci. Pewnym sposobem na oszukanie interpretera jest instrukcja POKE 56,208 lub też zredukowanie nazwy programu do jednego znaku.
3. Projektanci nie przewidywali dołączenia do Commodore 64 dwóch magnetofonów, stąd zmiana numeru urządzenia 1 jest niemożliwa. Numery 8 i 9 odnoszą się do stacji i mogą być zmieniane dowolnie, w zależności od potrzeb. Najczęściej spotyka się cztery numery stacji: 8, 9, 10 i 11.

[...] Mam C-64 i kilka pytań dotyczących tego komputera: 1. Dlaczego bez cartridge nie można grać?
2. Dlaczego, gdy kupuję kasety z grami, nie wgrywają mi się niektóre gry?
3. Czy obowiązkowo musi być monitor do tego komputera? [...]

1. Oczywiście, że można korzystać z gier na C-64 bez modułu. Jedyną różnicą będzie znacznie wydłużony czas wczytywania; dodatkowo będziesz musiał w większości wypadków wczytywać TURBO do pamięci komputera przed wczytaniem samej gry.
2. Kasety od handlarzy charakteryzują się dwoma „zaletami": po pierwsze są nagrywane byle jak w dużym pośpiechu, po drugie jasne jest, że sprzedawcy nie zależy na tym, abyś mógł zbyt dużo z takiej kasety wyciągnąć (bo możesz dać na przykład oprogramowanie kolegom i wtedy mniej osób kupi u niego). Głównym zaś felerem takich transakcji jest fakt zapisywania programów na różnych ustawieniach głowicy. Teoretycznie można głowicę ustawić samemu, lecz jest to na dłuższą metę niekorzystne dla Twojego magnetofonu. Na szczęście już niebawem takie transakcje będą zakazane prawem i za sprzedaż kradzionego oprogramowania można będzie wylądować w dobrze odosobnionym miejscu.
3. Oczywiście dobrze jest jak z komputerem współpracuje monitor (ostrzejszy i wyraźniejszy obraz oraz dźwięk) lecz komputer z równym powodzeniem można podłączyć do dobrego telewizora. Wyjątek mogą stanowić niektóre odbiorniki TV produkcji radzieckiej. Pamiętaj ponadto, że na świecie istnieje kilka systemów telewizyjnych, co oznacza, że nie wszystkie komputery muszą pasować do systemów popularnych w Polsce (SECAM i PAL).

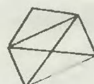
[...] Moja stacja dysków 1541 odczytuje programy zapisane na innych stacjach, lecz gdy ja sobie coś nagram na swojej stacji, to moi koledzy tego na swoich urządzeniach nie odczytają. Ich stacje w ogóle nie odczytują programów nagranych na mojej stacji, zachowują się tak, jakby dyskietka nie była sformatowana. Co jest tego powodem? [...]

Powodem może być rozstrojenie się, zmiana prędkości obrotowej, brudna i/lub uszkodzona głowica (głowice), niesprawna elektronika. Najlepszym rozwiązaniem będzie zaniesienie stacji do regulacji; dobrze byłoby, gdyby dokonał tego warsztat wyspecjalizowany.

* * *

Czy istnieje wersja programu CorelDRAW! na Amigę?

Niestety nie. Istnieją jednak dla niej inne programy wektorowe (structured drawing programs). Warto tu wymienić trzy z nich: ProVector 2.1 (Stylus Software), ProDraw 2.1 (Gold Disk) oraz najprostszy — Design Works 1.0 (New Horizons Software). Dwa pierwsze są programami na bardzo wysokim poziomie, dostosowanymi do współpracy z systemami DTP, takimi jak PageStream, ProPage czy Saxon Publisher. Umożliwiają one oczywiście pracę w kolorze, pracują w języku Postscript — słowem, stanowią niezłą alternatywę dla CorelDRAW!

LISTY  LISTY  LISTY  LISTY

Czy po podłączeniu do Amigi polskiego monitora Neptun 156 będą działały wszystkie programy?

Tak, Amiga zupełnie nie zwraca bowiem uwagi, jaki monitor jest do niej podłączony, ani przez które gniazdo — generuje po prostu sygnał wizji. Również dzięki temu możliwe jest uzyskanie wszystkich rozdzielczości na dowolnym monitorze.

Czy zgodność programowa pomiędzy Amigą 500 i 2000 jest stuprocentowa?

Różnice pomiędzy Amigą 2000 a Amigą 500 są tak minimalne, że można zaryzykować stwierdzenie, że tak. W tej pierwszej dodano bowiem tylko sloty rozszerzeń i układ je obsługujący — wszystkie pozostałe układy są takie same, jak w Amidze 500. Należy jednak pamiętać, że po wyposażeniu Amigi 2000 w różnej maści rozszerzenia, zachowuje się ona już nieco inaczej. Na przykład po zainstalowaniu „dopalacza" mogą przestać działać niektóre kiepsko napisane dema lub gry (pętle opóźniające wykorzystujące procesor wykonywane są dużo szybciej).

Jak sformatować dyskietkę na więcej niż 880 KB?

Niestety, nieprzerobiona Amiga nie potrafi zapisać na dyskietce więcej niż 880—900 KB. Przyczyna tego stanu rzeczy jest bardzo prosta: sterownik stacji dysków nie potrafi przesłać danych z większą niż standardowa szybkością, a co za tym idzie, nie jest możliwe uzyskanie większej gęstości zapisu. Są oczywiście inne sposoby na zwiększenie pojemności dyskietki, na przykład użycie ścieżek powyżej osiemdziesiątej, nikt jednak tego na Amidze nie stosuje.
Istnieją jednak przystawki zmieniające pracę sterownika Amigi, dzięki którym możliwe jest zapisanie danych z większą gęstością. Dobrym przykładem może tu być Sybil, która (we współpracy z dołączonym do niej programem) umożliwia zapisanie na zwykłej dyskietce 1.12 MB danych. Przystawka ta ma jeszcze cały szereg bardzo ciekawych funkcji, ale to już temat na oddzielny artykuł.

Czy freezer Action Replay II istnieje w wersji dla Amigi 2000?

Tak — to oczywiście moja wina, że nie wspomniałem o tym w tekście. Wersji Action Replay II przeznaczonej dla Amigi 2000 nie podłącza się jednak do złącza procesora, ma ona postać karty, instalowanej w jednym ze slotów wewnątrz komputera. Na zewnątrz trzeba wyprowadzić tylko przewód zakończony niewielkim pudełeczkiem, zawierającym wszelkie przełączniki służące do sterowania freezerem. Dla Amigi 2000 istnieje też najnowsza wersja Action Replay, oznaczona numerkiem trzy.

Czy można kupować Amigę 500 z myślą o profesjonalnych zastosowaniach, oczywiście po jej rozbudowie?

Na upartego możnaby stwierdzić, że tak, ja jednak radziłbym rozważenie kupna Amigi 3000 albo 2000. Kieruję się kilkoma powodami. Po pierwsze, sama konstrukcja A500 (wszystko w jednej obudowie, zasilacz walający się po biurku, brak możliwości ustawienia monitora na komputerze) jest wręcz zaprzeczeniem tego, czego można by oczekiwać po profesjonalnym systemie. Po drugie i najważniejsze: Amiga 500 ma minimalne możliwości rozbudowy, które kończą się właściwie po podłączeniu twardego dysku i rozszerzenia pamięci. O ile dysk twardy i rozszerzenie są porównywalne z tymi dla Amigi 2000, to jednak karty przyspieszające pracę są dla A500 znacznie uboższe. Jednak w ogóle są, w przeciwieństwie, np. do kart graficznych, jak Video Toaster, Impact Vision, Firecracker24 czy wreszcie DMI Resolver, które dostępne są jedynie dla posiadaczy Amig 2000 i 3000. System oparty na Amidze 500 będzie więc po prostu ubogi i do profesjonalnych zastosowań graficznych nie nadaje się zupełnie.

Jak zainstalować AmigaDOS na pustej dyskietce?

Od razu trzeba wyjaśnić, że AmigaDOS znajduje się w ROM, a na dyskietce instaluje się program inicjalizujący. Zawarty jest on w tzw. bootblocku, czyli dwóch sektorach odczytywanych natychmiast po włożeniu dyskietki do kieszeni stacji w chwili incjalizacji komputera. Aby pojawiło się więc okienko „AmigaDOS", wystarczy ów bootblock zainstalować za pomocą programu „Install" znajdującego się na dyskietce dołączanej do Amigi. Wpisz w CLI (lub Shell) „Install?" a następnie po włożeniu dyskietki podaj symbol stacji, w której znajduje się dyskietka („DFO:" dla stacji wewnętrznej, „DF1-3:" dla stacji zewnętrznych). Trzeba jednak pamiętać, że taka dyskietka będzie jedynie inicjalizowała system, nie będzie jednak zawierała wielu niezbędnych plików systemowych. Zainteresowanych odsyłam do artykułu „Co, gdzie, kiedy, czyli rozprawka o katalogach systemowych" (C&A, numer 02/92).

Czy Amiga wyposażona jest w gniazdo SCSI umożliwiające podłączenie CD-ROM od IBM PC?

Standardowo tylko Amigi 3000 i 3000 Tower wyposażone są w sterownik SCSI. Do wszystkich innych modeli Amigi, nie wyłączając 500 ani 1000, dostępnych jest jednak kilkadziesiąt modeli sterowników SCSI. I Amiga 3000, i pozostałe modele wyposażone w układ sterownika, dają możliwość podłączenia dowolnego urządzenia SCSI, czyli np. streamera, CD-ROM (także w wersji pozwalającej na zapis na dysku optycznym), dowolnego twardego dysku. Ważny jest tylko i wyłącznie standard, nie ma więc znaczenia, czy urządzenie (w tym przypadku CD-ROM) używane jest z IBM PC, Macintoshem, czy Amigą.

Czy sprzętowy emulator IBM PC ATonce wykorzystuje wbudowaną stację Amigi, czy też trzeba dokupić zewnętrzną stację 5.25 cala?

Wszystkie modele ATonce w pełni wykorzystują wbudowaną stację dyskietek 3.5". Po uruchomieniu emulatora, stacja pracuje w formacie IBM PC (osiemdziesiąt ścieżek, dwie strony, 720 KB), który potrafi zapisywać i odczytywać dowolna stacja 3.5 cala w dowolnym „pececie".

PIOTR LISZEWSKI
ANDRZEJ BOBEK

C-64 kontra ATARI 800XL

Od dawna wiadomo, że użytkownicy Atari 800 XL i Commodore 64 nie pałają do siebie zbytnią miłością i usiłują za wszelką cenę rozstrzygnąć znany dylemat: który z tych dwóch komputerów jest lepszy. Problem ten wcale nie jest łatwy do rozstrzygnięcia, jako, że suche przedstawienie danych technicznych nic nie da początkującym, natomiast upodobania są sprawą gustu. Ponieważ temat wydaje mi się dość istotny zapraszam wszystkich do dyskusji a poniżej przedstawiam swój pogląd na sprawę.

C-64 i Atari 800 XL są konstrukcjami przestarzałymi, ośmiobitowymi. Co utrzymuje je przy życiu? Tylko cena, bardzo duża liczba programów i tłum fanatyków. Z powodu niskiej ceny komputery te są konkurencją dla innych, większych i nowoczesnych, ale i droższych.

PROCESOR

Atari wyposażone jest w procesor 6502C i pracuje z zegarem o częstotliwości 1 MHz. Starsze wersje C-64 oparto na procesorze 6510, nowsze zaś na 8510. Obie wersje C-64 są oczywiście całkowicie zgodne ze sobą. Commodore 64 jest taktowane zegarem o częstotliwości 1 MHz.

PAMIĘĆ

BASIC i system operacyjny Atari mieszczą się w 24 KB pamięci ROM; Atari wyposażono w 64 KB pamięci RAM. Commodore 64 ma również 64 KB pamięci RAM, natomiast system operacyjny i interpreter BASIC — 20 KB (24 KB z generatorem znaków).

PORTY

Oba komputery są wyposażone w szereg gniazd umożliwiających podłączenie urządzeń zewnętrznych. Są to: gniazdo TV, gniazdo monitora (Audio-Video), port magnetofonu, stacji dysków, drukarki, porty joysticków i gniazda rozszerzające. C-64 jest wyposażony dodatkowo w port użytkownika zawierający nieco okrojoną wersję standardu RS-232 (m.in. napięcia są na poziomie TTL).

Do Atari można, oprócz wymienionych wyżej urządzeń, podłączyć także myszkę, tabliczkę graficzną i modem, do C-64 zaś pióro świetlne oraz myszkę i modem. Oba komputery są w stanie obsługiwać również tzw. „wiosełka" (kontroler potencjometryczny).

GRAFIKA I DŹWIĘK

Rozdzielczość w trybie graficznych Atari 800 XL wynosi 320∗192 a Commodore 64 — 320∗200 punktów. C-64 może wyświetlić na ekranie 25 wierszy po 40 znaków, a Atari — 24 takie wiersze. Jeśli chodzi o kolory, to C-64 wypada skromniutko — jest on w stanie generować maksymalnie 16 kolorów, Atari zaś aż 256.

Układ dźwiękowy C-64 jest 3-kanałowy; każdy kanał ma skalę 9. oktaw. Atari dysponuje 4 kanałami po 3.5 oktawy. Dla obu komputerów istnieje specjalizowany syntezer mowy — SAM — w wersjach polskiej, niemieckiej i angielskiej.

W Commodore 64 programowanie zarówno grafiki, jak i muzyki jest bardzo skomplikowane i żmudne ze względu na brak odpowiednich poleceń w BASIC. Można temu zaradzić korzystając z odpowiednich programów. Należą do nich: GEOPAINT, ART STUDIO, KOALA PAINTER, edytor muzyczny VOICE-TRACKER 4.0 i wiele innych.

PAMIĘCI ZEWNĘTRZNE

Oba komputery mogą korzystać z pamięci masowych w postaci magnetofonu i/lub stacji dysków. Niestety żadne z tych rozwiązań nie jest pozbawione wad. Współpraca C-64 z magnetofonem jest nieco nużąca ze względu na długi czas wczytywania programów, obsługa jest za to bardzo prosta. Istniejące oprogramowanie (myślę tu o TURBO) jest w stanie przyspieszyć transmisję w sposób programowy nawet dziesięciokrotnie.

W przypadku Atari sprawa się komplikuje. Długie wczytywanie programów w standardowym systemie jest tak samo nużące i irytujące jak w C-64. Tutaj jednak, aby przyspieszyć proces wczytywania, konieczna jest także przeróbka magnetofonu. Może jej dokonać tylko fachowiec. Problem pojawia się także przy wyborze odpowiedniego systemu; istnieje bowiem ich kilka, ale niestety niekompatybilnych ze sobą. Użytkownik C-64 nie ma takich problemów; choć systemów TURBO jest kilka, to jednak w olbrzymiej większości są one ze sobą zgodne, co zapewnia niemal nieograniczony dostęp do oprogramowania.

Najkorzystniejszym wyjściem jest, pomimo dość wysokiej ceny, zakup stacji dysków. Jest to urządzenie niezwykle pożyteczne, pozwalające wczytywać nawet bardzo długie i skomplikowane programy, w o wiele krótszym czasie niż z magnetofonu. Najbardziej znane stacje dysków Commodore i Atari pracują w oparciu o dyskietki średnicy 5.25". Pojemność dyskietki zapisanej przez stację Commodore 1541 wynosi 170 KB (1571 formatuje dyskietki na 340 KB!), a Atari 1050 — 127 KB. Stacje te są, w porównaniu z innymi, dość powolne, opracowano więc szereg metod sprzętowych i programowych do przyspieszenia ich pracy. W celu przyspieszenia pracy stacji 1541 można wykorzystać takie programy jak typu HYPRA LOAD, TURBO DISK, TURBO COU, SUPER DOS. W takie opcje są również wyposażone specjalne moduły takie jak FINAL II, III, ACTION REPLAY i podobne. Do tych samych celów służą przeznaczone dla 1050 programy US DOUBLER, TOP DRIVE 1050, HARPY 1050 czy też polski, uniwersalny TOMS TURBO.

Nie ma problemów także z kopiowaniem programów dyskowych. Istnieje, zarówno na Atari jak i na C-64, dość duża liczba programów kopiujących, służących do kopiowania pojedynczych plików, bądź całych dyskietek.

JĘZYKI PROGRAMOWANIA

Możliwości i szybkość C-64 oraz Atari 800 XL można zwiększyć poprzez zastosowanie interpreterów i kompilatorów. Najbardziej znane na Atari to: LOGO, TURBO BASIC XL, FORTH, PASCAL, C i oczywiście ACTION!, który jest przeznaczony wyłącznie na Atari, bardzo nowoczesny i jest jednym z najszybszych. Z kolei na C-64 dostępne są: SIMON'S BASIC, FORTH, LOGO, PASCAL, PILOT, C i znakomity WARSAW BASIC o bardzo dużych możliwościach ukierunkowanych na programowanie strukturalne. Warto wspomnieć, że samych dialektów języka BASIC jest kilkanaście.

OPROGRAMOWANIE

Chyba każdy znajdzie coś dla siebie w bogatym oprogramowaniu omawianych komputerów. Oferta jest naprawdę przebogata: gry wszelkiego typu (symulacyjne, strategiczne, logiczne, bij-zabij itp.), bazy danych, arkusze kalkulacyjne, edytory tekstu, edytory muzyczne i graficzne, programy edukacyjne.

Na koniec ważna uwaga: „Ataryki", pomimo najszczerszych chęci ich użytkowników, dogorywają. Konkurencja C-64 okazała się zbyt wielka — nawiasem mówiąc, jest to obecnie jedyny, produkowany komputer ośmiobitowy.

ŁUKASZ ADAMCZYK

OD REDAKCJI:

Artykuł Łukasza siłą rzeczy nie wyczerpał tematu; nie było na to po prostu miejsca. Tych, którzy chcieliby zabrać głos w tej dyskusji prosimy o listy, przy czym — uwaga — interesują nas Wasze opinie na temat PUNKTÓW WIDZENIA a nie „niezbite dowody wyższości" jednego komputera nad drugim.

JAK OKIEŁZNAĆ DRUKARKĘ: DOWNLOAD, DRUKOWANIE i POLSKIE LITERY

W poprzednim odcinku zajmowaliśmy się elementarnym sposobem konfigurowania drukarki — poprzez odpowiednie ustawienie mikrowyłączników (DIP-SWITCHES). Ich ustawienie ma kluczowe znaczenie dla pracy Twojej drukarki, lecz to jeszcze nie wszystko.

Od niepamiętnych czasów istnieje na całym świecie problem „własnych" znaków. W Polsce trudności te są doskonale znane, ze względu na „ogoniastość" naszego alfabetu. Z punktu widzenia producentów sprawa była na tyle istotna, że zaczęto jej z biegiem czasu poświęcać coraz więcej uwagi. Dzisiaj chyba wszystkie drukarki są firmowo wyposażane w opcję **download** dzięki czemu użytkownik może zaprogramować dla swoich potrzeb dowolny zestaw znaków. Zainteresowanych odsyłam do artykułu „Polskie znaki dla drukarki LC-20" w tym numerze „C&A".

Od strony teorii „operacja DOWNLOAD" wygląda następująco: każda drukarka jest wyposażona w pamięć ROM (lub EPROM zawierającą kilka lub kilkanaście zestawów znaków i program sterujący pracą urządzenia. Dołączana jest również pamięć RAM pełniąca rolę bufora do którego są wpisywane znaki do wydrukowania. Pamięć tę można jednak wykorzystać inaczej: po wydaniu odpowiedniego polecenia możliwe jest skopiowanie zawartości ROM do pamięci RAM drukarki. Następnie wyłączamy ROM i korzystamy ze skopiowanego zestawu znaków znajdującego się teraz w pamięci RAM. Zaleta takiego rozwiązania jest oczywista — zawartość pamięci RAM czyli zestawu znaków można modyfikować!

W ten sposób użytkownik dostaje do ręki narzędzie o bardzo dużych możliwościach. Po pierwsze: możesz utworzyć setki własnych krojów pisma, znaków graficznych i symboli. Po drugie: nie tracisz na szybkości druku. Po trzecie: możliwości te są niezależne od komputera. Nie jest ważne czy masz peceta, C-16 czy Amigę HIGH-END — każdy z tych komputerów daje Ci identyczne możliwości sterowania drukarką.

Skoro wszystko jest take piękne, to dlaczego wszędzie jest jakoś głucho na temat tej metody? Odpowiedź jest prosta: ponieważ programów korzystających z tego sposobu jest stosunkowo niewiele. Polskie literki na wydruku można uzyskać na kilka sposobów, przy czym każdy z nich ma swoje wady i zalety. Największą zaletą jaką daje Ci download jest szybkość, są jednak tacy, którym bardziej zależy na jakości. Download załatwia sprawę dla znaków o określonych wymiarach, jeśli jednak potrzebujesz symbolu o wysokości kilku liter, będziesz musiał stosować sztuczki, np. cofanie papieru.

Jakość oprogramowania wyznacza jednocześnie grupę jego użytkowników. Za przykład może posłużyć edytor CHIWRITER znany szeroko z „pecetów". Jego niezwykła popularność w Polsce jest związana między innymi z tym, że użytkownik NIE MUSI MYŚLEĆ jak uzyskać polskie znaki. Wystarczy wiedzieć, który klawisz funkcyjny wcisnąć i na ekranie (a także na wydruku) pojawia się polska literka. Proszę zwrócić uwagę, że z tego samego powodu rozpowszechnił się wśród użytkowników C-128 (i C-64) TEXTOMAT — edytor stosunkowo wolny i mało elastyczny, miał jednak zdefiniowany (przez giełdę) zestaw polskich znaków oraz PRYMITYWNIE PROSTY układ ich uzyskiwania.

Edytory tekstu działające w oparciu o download są w stanie zapewnić taki komfort pod warunkiem, że twórca programu o tym pomyśli. W większości zagranicznych edytorów taka potrzeba nie istniała, gdyż narodowy zestaw znaków „ogonków" nie zawierał; bywało również i tak, że na niektóre kraje (np. Niemcy) producent opracowywał oddzielną wersję programu. W zamian pojawił się w Polsce popyt na edytory specjalizowane, to znaczy takie, dzięki którym można było uzyskać nie tylko polskie litery ale również symbole matematyczne, wzory chemiczne, cyrylicę i setki innych znaków. Do nich należą między innymi CHIWRITER, TEXTOMAT, FONTMASTER i inne.

Aby zrozumieć wady i zalety wynikające ze stosowania (bądź niestosowania) nie tylko downloadu, lecz również innych metod pozwalających na uzyskanie polskich liter, muszę nieco odbiec od tematu. Jak wiadomo drukarka może pracować w dwóch trybach, tekstowym i graficznym; oba pozwalają na uzyskanie polskich liter lecz w każdym dzieje się to inaczej.

W tym pierwszym tekst do wydrukowania jest traktowany jako zestaw symboli (rozpoznawanych na podstawie przesyłanego do drukarki kodu znaku). W trybie graficznym natomiast drukarka otrzymuje nie kod znaku, lecz ciąg bitów określających kształt znaku. Pierwsza metoda charakteryzuje się dużą szybkością działania, druga zaś — możliwością generowania znaków o dowolnych kształtach. Połączenie obu tych zalet nosi nazwę „drukarka laserowa".

Ogólnie istnieją trzy metody uzyskiwania polskich znaków w wydruku:

1. Poprzez przełączanie drukarki do pracy w trybie graficznym na czas wydruku niestandardowego znaku lub też potraktowania całego tekstu jako OBRAZU GRAFICZNEGO. Tak działa na przykład program FONTMASTER II, PRINTFOX czy TEXTOMAT 64/128. Potraktowanie całej strony tekstu jako obrazu graficznego pozwala zwykle na bezproblemowe łączenie tekstu z grafiką (i w taką opcję jest wyposażony np. TEXTOMAT) i wydruk znaków o niestandardowych wymiarach. Wadą jest powolny wydruk; do zalet należy zaliczyć bardzo wysoką jakość wydruku. Największe katusze cierpią tu właściciele Amig i drukarek 24.igłowych; brak w niektórych edytorach odpowiednich programów sterujących drukarką w trybie graficznym bardzo ogranicza jakość wydruku...
2. Download. Przed wydrukiem edytor przesyła do drukarki serię kodów sterujących powodujących przeniesienie wybranego zestawu znaków do pamięci RAM drukarki i zmieniających cały zestaw lub pojedyńcze znaki. Typowym przykładem może tu być program o nazwie EDYTOR PL. Rozwiązanie to ma jednak inne ograniczenia: wydruk jest już szybki, lecz użytkownik dysponuje ograniczonym zestawem znaków. Istnieje oczywiście możliwość zmiany zestawu w trakcie druku, jest to jednak czasochłonne i rzadko stosowane. Po przypadkowym wyłączeniu drukarki trzeba przesłać zestaw znaków do drukarki na nowo. Zwykle stosuje się download wraz z konkretnym programem.
3. Trzecia metoda, to wymiana oryginalnej pamięci ROM i zainstalowanie kości z zestawem polskich znaków. Jest to sposób najwygodniejszy, jednakże w ten sposób użytkownik jest pozbawiony na stałe kilkunastu symboli. Oczywiście może to nie mieć znaczenia w sytuacji, gdy drukarka będzie wykorzystywana do ściśle określonego celu — np. adresowania kopert czy produkcji zawiadomień. Problemy powstają zwykle w momencie, gdy potrzebna jest jakaś tabelka lub wydruk zawierający znaki przedefiniowane na polskie literki.

Za miesiąc przedstawię najpopularniejsze programy służące do edycji tekstu, będą też opisy niektórych z nich.

KLAUDIUSZ DYBOWSKI

SUPERMARKET

■ Sprzedam C-64, magnetofon (gwarancja do 1993), joystick QSII Plus, FINAL III, 300 programów, literaturę. CENA: 2 mln. T. Hatylak, Kochanowskiego 15/6, 38-500 SANOK, tel. 30-602

■ Pilnie sprzedam C-64 VGS (stan idealny), magnetofon, BLACK BOX, FINAL III, 420 programów. Literatura i joystick gratis. Informacje: przyślij kopertę + znaczek. Krzysztof Piętak, ul. Ludowa 15, 32-547 Jaworzno 4, woj. katowickie.

■ Sprzedam C-64 z magnetofonem, monitorem, stacją dysków 1541-II, 2 joystickami, kasetami i dyskietkami, Jacek Łucki, Oś. J. Krasickiego 4/42, 28-100 BUSKO-ZDRÓJ, woj. kieleckie (telefon: 71-93).

■ Sprzedam C-64, magnetofon, moduł BLACK BOX, joystick, 16 kaset z programami. CENA: 1.5 mln. Paweł Kocoń, ul. Kniewskiego 8/9, Oś. Syberka, 42-500 BĘDZIN.

■ Sprzedam C-64 II, magnetofon, moduł BLACK BOX, 40 kaset, literaturę. CENA: 3.1 mln. Sebastian Skowronek, ul. Gałczyńskiego 10/2, 08-300 SOKOŁÓW PODLASKI.

■ Sprzedam C-64 II z magnetofonem (na gwarancji do sierpnia 1992). CENA: 1.6 - 1.7 mln. Sergiusz Juraszek, ul. Zawiśle 18, 43-430 SKOCZÓW, tel. 34-29.

■ Sprzedam C-64 II, magnetofon, joystick i programy w idealnym stanie. CENA: 1.7 mln do 31.08.92, potem 2 mln. Arek Grzywna, ul. Krasickiego 58/52, 23-400 BIŁGORAJ.

■ Sprzedam (lub zamienię na Amigę 500) Commodore 64 II, magnetofon, literaturę, 500 gier, 2 moduły. G. Zrebny, ul. Rataja 31/5, 42-400 ZAWIERCIE.

■ Sprzedam Commodore 64 z magnetofonem (gwarancja), moduł BLACK BOX, 2 joysticki. CENA: 2 mln. Dariusz Kliczner, Oś. ZWM 9/14, 32-540 TRZEBINIA.

■ Zamienię nowy Commodore 64 VGS (gwarancja), magnetofon 1530, 4 moduły, joystick i literaturę na mało używany Commodore 128 lub CPC 6128 MONO (ewentualnie dopłacę). R. Kaczmarczyk, ul. Szopena 48c m2, 41-400 MYSŁOWICE, woj. katowickie.

■ Sprzedam Amigę 500, 1 MB RAM, 320 dyskietek, oprzyrządowanie. CENA: 13 mln. Rafał Roszak, SZCZECIN, tel. 50-14-12.

■ Kupię instrukcje do gier dla Amigi 500: Carthage, Elite, Joan of Argy, Megalomania, Maupiti Islands, Oil Imperium, Ports of Call, Railroad Tycoon, Vermeer Society, Columbus. Marek Grabowski, B. Chrobrego 61/2, 65-052 ZIELONA GÓRA.

■ Zamienię colorfon C-230B, Datassette, 500 gier (kasety), joystick, BLACK BOX III, obszerną literaturę, na używaną stację dysków do C-64. Janusz Jarema, Kwiatowa 4, 11-300 BISKUPIEC, tel. (15) 30-61.

■ THE AMIGANTS CLUB. Wymiana doświadczeń i oprogramowania, porady (przyślij kopertę zwrotną ze znaczkiem). Ulica Kuflewska 4/22, 03-554 WARSZAWA, tel. 679-17-25.

SUPERMARKET 06/92

SUPERMARKET 06/92

SUPERMARKET 06/92



Odcinek dla Poczty	Odcinek dla posiadacza rachunku	Potwierdzenie dla wpłacającego
Zł	Zł	Zł ....
słownie złotych	słownie złotych	słownie złotych
wpłacający:	wpłacający:	wpłacający: ....
Dokładny adres i kod	Dokładny adres i kod	Dokładny adres i kod ....
Wydawnictwo BAJTEK ul. Wspólna 61 00-687 Warszawa Bank Agrobank S.A. 470005-1834-131 ul.Grochowska 262, 04-398 Warszawa	Wydawnictwo BAJTEK ul. Wspólna 61 00-687 Warszawa Bank Agrobank S.A. 470005-1834-131 ul.Grochowska 262, 04-398 Warszawa	Wydawnictwo BAJTEK ul. Wspólna 61 00-687 Warszawa Bank Agrobank S.A. 470005-1834-131 ul.Grochowska 262, 04-398 Warszawa
Datownik Oplata podpis przyjmującego	Datownik **WYPEŁNIJ TEN KUPON NA ODWROCIE**	Datownik Oplata podpis przyjmującego

NINIEJSZYM ZAMAWIAM:

☐ DYSKIETKĘ 3.5" dla Amigi, sztuk:........, cena 21000 zł

☐ DYSKIETKĘ 5.25" dla Amigi, sztuk:........,cena 21000 zł

☐ KASETĘ dla Commodore 64, sztuk:........, cena 100000 zł

DO ZAMÓWIENIA DOŁĄCZAM KOPIĘ DOWODU WPŁATY

IMIĘ:..

NAZWISKO: ....

ULICA:..

MIASTO: ....

UWAGI:...

DYSKIETKI 3.5" I 5.25" DLA AMIGI

Choć o różnych średnicach obie zawierają te same pliki. Są to:

■ trzy pliki pozwalające na uzyskanie polskich liter na drukarkach Star serii LC - 10/20/200 (download),

■ specjalny moduł (sample) przedstawiający możliwości ulepszonego samplera, którego schemat opublikowaliśmy w "C&A" 04/92. Schemat nowej wersji zostanie opublikowany w numerze wrześniowym "C&A".

■ Pozostałe miejsce zajmują moduły programu Protracker, oraz kilka plików tekstowych, a wśród nich (między innymi) prawa Murphy'ego (w języku angielskim) oraz ciekawostka z uniwersyteckiej sieci amerykańskiej.

KASETA DLA COMMODORE 64

Kaseta ta zawiera **wszystkie programy opublikowane dotychczas w naszym miesięczniku - od numeru 01/92 do numeru 06/92 WŁĄCZNIE.**

Jak zapewne wiesz, w "C&A" drukowaliśmy programy nie tylko dla magnetofonu, lecz także dla stacji dysków; również i one znajdują się na tej kasecie. Pamiętaj jednak, że niektóre z nich nie będą poprawnie współpracować z magnetofonem i wymagają stacji dysków w zestawie.

Aby wyeliminować problemy z głowicą związane z jakością nośnika, stosujemy wyłącznie kasety **FIRMOWE** (FUJI, BASF). Wolne miejsce wypełniają programy PD i "demka". Kaseta jest nagrana **JEDNOSTRONNIE.**

!!! UWAGA !!!

Warunkiem realizacji zamówienia jest PRAWIDŁOWE wypełnienie przekazu wpłaty. W wypadku wypełnienia przekazu, niezgodnie z poniższymi danymi **REDAKCJA NIE BIERZE ODPOWIEDZIALNOŚCI ZA STRATY PONIESIONE Z TEGO TYTUŁU, A ZAMÓWIENIA TAKIE NIE BĘDĄ REALIZOWANE.** Aby uniknąć pomyłki skorzystaj z blankietu zamieszczonego poniżej.

Właściciel konta:
WYDAWNICTWO "BAJTEK", 00-687 Warszawa, ul. Wspólna 61

Bank :
Bank Agrobank S.A., ul Grochowska 262, 04-398 Warszawa

Nr konta: 470005-1834-131.



Kupon ważny do 06.07.1992

ZAMAWIAM PRENUMERATĘ:

( ) MIESIĘCZNIKA "C&A"
( ) MIESIĘCZNIKA "BAJTEK"
( ) DWUMIESIĘCZNIKA "TOP SECRET"

ZAMAWIAM NUMERY "C&A":

* 01/92: () magazyn () dyskietka
* 02/92: () magazyn () dyskietka
* 03/92: () magazyn () dyskietka
* 04/92: () magazyn () dyskietka
* 05/92: () magazyn () dyskietka

 Uwagi:

Kupon ważny do 06.07.1992

ZAMAWIAM PRENUMERATĘ:

( ) MIESIĘCZNIKA "C&A"
( ) MIESIĘCZNIKA "BAJTEK"
( ) DWUMIESIĘCZNIKA "TOP SECRET"

ZAMAWIAM NUMERY "C&A":

* 01/92: () magazyn () dyskietka
* 02/92: () magazyn () dyskietka
* 03/92: () magazyn () dyskietka
* 04/92: () magazyn () dyskietka
* 05/92: () magazyn () dyskietka

 Uwagi:

Na wszystkich odcinkach blankietu należy wpisać CZYTELNIE atramentem, długopisem lub pismem maszynowym kwotę cyframi oraz słownie, imię i nazwisko oraz adres wpłacającego.



LISTY  LISTY  LISTY  LISTY

Czy monitor kolorowy PHILIPS CM8833–II może pracować w trybie „interlace"? Jeśli tak to jak uzyskać ten tryb?

Mam problemy z użyciem muzyki napisanej pod „Soundtrackerem"...

Sławomir Goliasz, Bielsko-Biała

Każdy popularny monitor powinien współpracować z trybem interlace, niektóre nawet mogą usuwać charakterystyczne „drżenie" ekranu (monitory typu multisync), jednak tylko po zainstalowaniu urządzenia typu flicker-fixer. Tryb interlace można uzyskać na wiele sposobów, najczęściej wybierając go przy konfiguracji programu (w Deluxe Paint na samym początku; można też interlace ustawić dla ekranu Workbencha używając programu Preferences). Tryb interlace wybiera się poprzez ustawienie bitu 1 (licząc od zera) w rejestrze sprzętowym BPLCON0 — adre $dff100.

Zwykle razem z programem typu tracker na dysku dołączana jest procedura w asemblerze umożliwiająca wykorzystanie napisanej przez siebie muzyki w innym programie. Podam teraz przykładowy sposób uruchomienia tej procedury jednakże potrzebny będzie dowolny asembler, najlepiej MasterSeka (wersja powyżej 1.70) lub AsmOne (uwaga! „najnowsza" wersja 1.21 jest w pełni działającą przeróbką! Proponuję poszukać wersji 1.01).

Po wgraniu np. AsmOne program zapyta się o typ i wielkość pamięci roboczej — wybieramy „c" (chip) i ok. 300 KB (zależy od długości naszej muzyczki). Następnie wczytujemy procedurę odtwarzającą moduł zleceniem „r" podając potem jej nazwę (nie trzeba dodawać przyrostka „.s" na końcu — zrobi to za nas program). Gdy to się powiedzie należy wejść do wbudowanego edytora tekstu programu i na samym początku wpisać następujące rozkazy:

```
        bsr.s mt_init          ; inicjalizacja procedury
petla:  cmp.b #$80,$dff006     ; czekanie na linię nr. 128
        bne.s petla
        move.w #$f00,$dff180   ; zmiana koloru ekranu
        bsr.s mt_music         ; odegranie jednej nuty
        move.w #$000,$dff180   ; zmiana koloru ekranu
        btst #10,$dff016       ; test prawego guzika myszy
        bne.s petla            ; jeżeli nie wciśnięty to
                               ; gramy dalej
        bsr.s mt_end           ; czynności końcowe
        rts                    ; koniec programu
```

Następnie należy odszukać ciąg znaków „mt_data" (wyjść z edytora, wydać zlecenie „t" oraz „lmt_data" i ponownie wejść do edytora). Kasujemy całą linię i piszemy:

mt_data: incbin „DF0:mod.coolmuzak"

Spowoduje to wczytanie modułu o nazwie „mod.coolmuzak" do naszego programu. Oczywiście do cudzysłowu można wpisać nazwę własnego dzieła.

Następną czynnością jest kompilacja naszego programu. Należy wyjść z edytora wpisać „a", a gdy program zapyta się o opcje (AsmOne tego nie robi) należy wcisnąć sam klawisz Return. Jeżeli wszystko poszło dobrze (komunikat „No errors") możemy wysłuchać (i obejrzeć!) swoje dzieło pisząc „j".

W przypadku wykrycia przez program błędu trzeba znaleźć jego źródło — najczęściej nie ma pliku podanego w po dyrektywie „incbin" na dysku. W innym przypadku może się okazać, że asembler nie lubi instrukcji „bsr.s mt_..." we fragmencie, który wpisaliśmy — należy wtedy wszystkie trzy takie odwołania zamienić na „jsr mt_...".

Aby zapisać nasz moduł jako wykonywalny program należy go skompilować, a następnie wydać zlecenie „woc" i podać nazwę pliku.

jrme

(...) Po przeczytaniu kolejnych numerów C&A doszedłem do wniosku, że na Amigę nie ma ŻADNYCH kompilatorów. (...) Można powiedzieć, że dość dobrze znam Basic na C=64 i Pascal oraz C++ na IBM, jednakże obawiam się, że dla Amigi nie ma tych języków. Gdzie można nabyć legalne kompilatory Pascala i C?

(...) Czy to prawda, że Amiga 500 Plus jest niekompatybilna z 500?

Tomasz Hatylak, Sanok

Pozory mylą, dla Amigi napisano wiele kompilatorów i interpreterów języków wyższego rzędu. Pascal, Forth, Fortran, Modula-2, C, Pilot, Cobol,... Basic — to tylko niektóre. Oczywiście niektóre z nich są nieznane z różnych powodów (głównie dlatego, że nikomu nie chce się ich sprowadzić do Polski). Najpopularniejsze języki, w których pisze się programy dla Amigi to bez wątpienia C (wspaniały pakiet Sas C), Pascal (Hispeed Pascal, zgodny z Turbo Pascalem) oraz Basic (perfekcyjny GFA-Basic czy AMOS — bogato rozszerzony Basic wykorzystujący wszystkie cechy sprzętowe Amigi). Modula-2 też ma swoich zwolenników (w niej napisano jeden z lepszych edytorów ANSI — LaDraw). Wiele kompilatorów to programy public domain (np. North C, Fortran) i należy ich szukać na giełdzie, natomiast rozbudowane pakiety przeznaczone dla profesjonalnych programistów są to programy bardzo drogie, jednakże dające pełnię władzy nad maszyną. Można je spotkać w wersji pirackiej jednak zaraz wychodzi na wierzch brak instrukcji, która zwykle liczy kilkaset stron.

Najprostszym sposobem nabycia kompilatora C czy Pascala jest wizyta dobrze zaopatrzonym sklepie np. w Niemczech — u nas takie programy (oryginalne) trzeba będzie zamawiać u pośredników, co pociąga za sobą dodatkowe koszty i oczekiwanie.

Niestety, główną ofertą dystrybutorów legalnego oprogramowania długo jeszcze pozostaną gry, choć zapewne już w najbliższej przyszłości będzie można nabyć interpreter i kompilator AMOSa.

Co do zgodności Amigi 500 Plus z modelem 500 to napisaliśmy o tym w teście A500+ zamieszczonym w jednym z ostatnich Bajtków — ogólnie mówiąc większość oprogramowania użytkowego „chodzi", jednakże z grami może być większy kłopot — bez problemów działa ok. 60%.

jrme

1. Czy A2000 jest produkowana?
2. Czy na dyskietce systemowej jest program obsługi modemu?
3. W drugim numerze C&A opublikowaliście test Action Replay Mk II (...), czy istnieje taki „freezer" do A2000 i czym różni się Mk II od Mk III?

Janusz Pankiewicz, Kłodzko

1. Amiga 2000 jest wciąż produkowana, jednak na jej bazie powstały dwa inne modele firmowane przez Commodore — Amiga 2500 (czyli A2000 + karta turbo) i Amiga 1500 (A2000 bez dysku twardego; w Europie model 2000 zazwyczaj jest sprzedawany razem z dyskiem twardym, do nas jednak trafiają tańsze modele). Wszystkie fabrycznie nowe komputery są wyposażone w układ Super Fat Agnus pozwalający na zamontowanie do 2MB CHIP-RAM na płycie.
2. Niestety, ani na dysku Workbench, ani na Extras nie ma programu terminala.
3. Posiadacze Amigi 2000 zmuszeni są do kupna specjalnej wersji modułów Action Replay przeznaczonych dla tego modelu, zwykle kosztuje to ok. 10 funtów angielskich więcej. Mk III jest ulepszeniem modułu Mk II, ma więcej przydatnych opcji oraz wbudowany program kopiujący, tzw. „Burst Nibbler". Opis Mk III ukazał się w numerze 6 pisma C&A

jrme